

Proyecto Fin de Grado

Grado en Ingeniería de Tecnologías Industriales

Integración de equipos en célula de fabricación flexible

Autor: Ramón Iglesias Bayo

Tutor: Dr. Luis Fernando Castaño Castaño

Dep. de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2016



Proyecto Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Integración de equipos en célula de fabricación flexible

Autor:

Ramón Iglesias Bayo

Tutor:

Luis Fernando Castaño Castaño

Profesor titular

Departamento de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016

Proyecto Fin de Grado: Integración de equipos en célula de fabricación flexible

Autor: Ramón Iglesias Bayo

Tutor: Luis Fernando Castaño Castaño

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2016

El Secretario del Tribunal

Agradecimientos

Este trabajo implica el final de una de las etapas de mi vida. Por ello, me gustaría agradecer a mis padres todo el apoyo que me han dado durante estos últimos cuatro años, dándome ánimos en aquellos momentos en los que más lo he necesitado.

También quiero agradecer a mi tutor Fernando Castaño todo el tiempo que me ha dedicado y todo lo que me ha enseñado. Sin él, la realización de este trabajo hubiese sido menos entretenida sin lugar a dudas.

Resumen

En el siguiente proyecto se ha realizado la integración de los equipos existentes en una célula de fabricación flexible ubicada en los laboratorios del Departamento de Sistemas y Automática de la Escuela Técnica Superior de Ingeniería (Universidad de Sevilla).

La célula de fabricación flexible consta de un sistema de transporte formado por 5 tramos de cintas transportadoras sobre las que se depositan bandejas. Estas bandejas son servidas o recogidas del sistema de transporte mediante un alimentador de bandejas. También se encuentra un sistema robotizado de almacenaje, que sirve o almacena pallets los cuales pueden contener piezas.

El sistema de transporte y el alimentador de bandejas poseen autómatas Modicon M340 de Schneider Electric. Se ha desarrollado el procedimiento para la conexión entre ellos por una red local Ethernet usando protocolo Modbus en Unity Pro XL. El sistema de transporte ejercerá de maestro en la comunicación, mientras que el alimentador de bandejas tomará el papel de esclavo.

El sistema robotizado de almacenaje está controlado por un Controlador-B de Eshed Robotec, el cual se encuentra conectado por puerto serie al autómata. Se ha desarrollado el procedimiento de conexión entre equipos en modo de caracteres en Unity Pro, en la cual se ha implementado un algoritmo de detección de errores de tipo CRC (Código de Redundancia Cíclica) en el envío y recepción de los mensajes.

La supervisión de la instalación se realiza mediante un sistema SCADA (sistema de supervisión, control y adquisición de datos) desarrollado en Vijeo Citect 7.0. Se exploran dos métodos distintos de comunicación con los autómatas: mediante OPC (OLE for Process Control) o mediante Unity Speedlink, conexión sólo disponible entre equipos de Schneider Electric. También se presenta una tabla comparativa entre estos métodos presentados.

Por último, se desarrolla una aplicación de ejemplo, en la que se usan los distintos métodos de conexionado entre los equipos. El objetivo de la aplicación es extraer pallets usando el sistema robótico de almacenaje para que sean procesados en uno de los puestos de trabajo de la célula de fabricación flexible. Tras el procesado, serán almacenado de nuevo en la zona habilitada para ello en el almacén. Las bandejas necesarias para el transporte de los pallets son proporcionadas por el alimentador, que ajusta el número de bandejas en circulación al número de peticiones de procesado. Esta aplicación demostrativa también posee una aplicación SCADA con el que se puede realizar la supervisión del sistema.

Agradecimientos	vii
Resumen	ix
Índice	xi
Índice de Tablas	xiii
Índice de Figuras	xv
Lista de abreviaturas	xix
1 Introducción	1
2 Objetivos	3
3 Descripción de la instalación	5
3.1. <i>Células de fabricación flexible</i>	5
3.2. <i>Descripción de la instalación utilizada</i>	5
3.2.1 Circuito de cintas transportadoras	6
3.2.2 Almacén matricial ASRS de Eshed Robotic	6
3.2.3 SCORBOT-ER 4u	8
3.2.4 Robot Sony SCARA	8
3.2.5 Alimentador de bandejas	8
3.2.6 Zona de clasificación	9
4 Comunicación entre el almacén robotizado de pallets y el PLC principal	11
4.1. <i>Conexión del hardware</i>	11
4.1.1 Conexión física	11
4.1.2 Configuración del hardware	13
4.2. <i>Comunicación maestro-esclavo</i>	15
4.2.1 Estructura del mensaje	16
4.3. <i>Algoritmos de redundancia</i>	17
4.3.1 Cálculo de CRC	18
4.3.2 Implementación de la división de polinomios	19
4.3.3 Creación de variable asociada al Puerto serie	19
4.3.4 Librería de bajo nivel usada	20
4.3.5 Creación de tipos para las tablas de gestión	21
4.3.6 Librería creada para la comunicación en Unity Pro XL	21
4.4. <i>Programación del almacén ASRS</i>	23
4.4.1 Librería de bajo nivel	23
4.4.2 Librería creada para la comunicación del almacén.	24
5 Comunicación entre autómatas	31
5.1. <i>Conexión del hardware</i>	31
5.2. <i>Configuración de la red Ethernet en autómata usando Unity Pro XL</i>	33
5.3. <i>Librería básica usada</i>	34
5.3.1 ADDM	34
5.3.2. READ_VAR	35

5.3.3. WRITE_VAR	35
6 Comunicación de autómatas con el SCADA	37
6.1. Conexión OPC	37
6.1.1 Procedimiento de conexión del autómata al SCADA	37
6.2. Unity Speedlink Static	45
6.2.1 Procedimiento de conexión del autómata usando Unity Speedlink Static	46
6.3. Uso de vectores	50
6.4. Tabla comparativa entre Unity Speedlink y OPC	51
7 Aplicación práctica: Célula de fabricación flexible	53
7.1. Sistema de transporte	53
7.1.1 Descripción del sistema de transporte	53
7.1.2 Guía GEMMA	57
7.1.3 Implementación usando Unity Pro XL	60
7.2. Alimentador de bandejas	69
7.2.1 Descripción de la máquina.	69
7.2.2 Presentación de la guía GEMMA desarrollada	72
7.2.3 Implementación usando Unity Pro XL	73
8 Aplicación práctica: Sistema de supervisión SCADA	79
8.1 Apariencia de la ventana	79
8.2 Pantallas principales	80
8.2.1 Home	80
8.2.2 Info instalación	81
8.2.3 Transporte	81
8.2.4 Almacén matricial	82
8.2.5 Alimentador de bandejas	83
8.2.6 Ayuda	84
8.3 Alarmas configuradas	84
8.4 Registro de actividad en archivo de texto	86
8.5 Identificación de usuarios	87
Anexo A: Programación del autómata principal	89
Anexo B: Programación del alimentador de bandejas	131
Anexo C: Programación del sistema robotizado de almacenaje	147
Referencias	175

Índice de Tablas

Tabla 1: Panel trasero del Controlador-B.	12
Tabla 2: Descripción de los elementos del módulo.	12
Tabla 3: Detalle de los pines del conector RJ45.	14
Tabla 4: detalles de los pines del conector DB9.	14
Tabla 5: Parámetros para la configuración del modo de caracteres.	14
Tabla 6: Estructura del mensaje de petición y de respuesta a la petición.	16
Tabla 7: Mensajes de peticiones.	16
Tabla 8: Mensajes posibles de respuesta del almacén.	17
Tabla 9: Operación de suma en el campo algebraico binario.	18
Tabla 10: Operación de multiplicación en el campo algebraico binario.	18
Tabla 11: Descripción de los parámetros de PRINT_CHAR.	21
Tabla 12: Descripción de los parámetros de INPUT_CHAR.	21
Tabla 13: Parámetros de entrada y salida de la función Envio_mensaje_con_CRC.	22
Tabla 14: Parámetros de entrada y salida de Lectura_mensaje_recibido_Con_CRC.	23
Tabla 15: Almacenamiento según su categoría.	24
Tabla 16: Parámetros de entrada y salida del programa CRC.	24
Tabla 17: Parámetros de entrada y salida del programa CRCEN.	25
Tabla 18: Parámetros de entrada y salida del programa ENVAL.	25
Tabla 19: Características técnicas del conmutador disponible en la instalación.	32
Tabla 20: Descripción de los puertos de la CPU BMX P34 2020.	32
Tabla 21: Parámetros de entrada y salida de READ_VAR.	35
Tabla 22: Parámetros de entrada y salida de WRITE_VAR.	35
Tabla 23: Sensores de las cintas transportadoras.	54
Tabla 24: Actuadores de la cinta transportadora.	55
Tabla 25: Entradas del autómata principal.	62
Tabla 26: Salidas del autómata principal.	63
Tabla 27: Descripción de los sensores.	70
Tabla 28: Descripción de los actuadores.	70
Tabla 29: Alarmas con las descripciones implementadas en el SCADA.	85

Índice de Figuras

Ilustración 1: Diferencias entre la variación del producto y de la cantidad para los distintos tipos de automatización [1].	5
Ilustración 2: Esquema general de la planta.	6
Ilustración 3: Almacén matricial ASRS.	7
Ilustración 4: Almacén matricial de la célula de fabricación flexible [2].	7
Ilustración 5: Puesto de trabajo del Scorbob con el dispensador de piezas y robot SCARA.	8
Ilustración 6: Dispensador de bandejas.	9
Ilustración 7: Panel trasero del Controlador-B [5].	11
Ilustración 8: Descripción física del módulo [6].	12
Ilustración 9: Correspondencia entre pines de los conectores [3].	13
Ilustración 10: Detalle del puerto RJ45 de los procesadores BMX P34 1000/2010/2020 [4].	13
Ilustración 11: Pantalla de configuración de la conexión en modo de caracteres.	14
Ilustración 12: Posibilidades en la comunicación.	15
Ilustración 13: Pasos para la creación de la variable asociada al canal.	20
Ilustración 14: Librería de bajo nivel usada en Unity Pro XL.	20
Ilustración 15: Bloques usados como librería básica.	22
Ilustración 16: Diagrama de flujo del programa RECEI.	26
Ilustración 17: Diagrama de flujo del programa CRC.	27
Ilustración 18: Diagrama de flujo del programa CRCEN.	28
Ilustración 19: Diagrama de flujo del programa ENVAL.	29
Ilustración 20: Ubicación y detalle del conmutador [11].	31
Ilustración 21: Esquema simplificado de la red local.	32
Ilustración 22: Ubicación del puerto Ethernet (3) en la CPU BMX P34 2020.	32
Ilustración 23: Ventana de configuración de nueva red Ethernet.	33
Ilustración 24: Ventana de configuración de la dirección IP de la red Ethernet.	33
Ilustración 25: Detalles de la configuración de la red Ethernet.	34
Ilustración 26: Bloques usados en la comunicación por MODBUS.	34
Ilustración 27: Exportación de variables en Unity Pro XL.	38
Ilustración 28 Creación del nuevo Alias.	38
Ilustración 29: Ventana de configuración de la dirección.	39
Ilustración 30 Configuración del Alias.	39

Ilustración 31: Elección del servidor OPC.	40
Ilustración 32: Creación del nuevo grupo de variables.	40
Ilustración 33: Ventana de creación de ítems.	41
Ilustración 34: Ventana en el caso de una conexión incorrecta.	41
Ilustración 35: Explorador de proyectos de Vijeo Citect.	42
Ilustración 36: Creación de servidor E/S.	42
Ilustración 37: Selección de tipo de dispositivo.	42
Ilustración 38: Selección del método de conexión.	43
Ilustración 39: Selección de la dirección del servidor OPC.	43
Ilustración 40: Opción de vincular a una base externa de datos de tags.	43
Ilustración 41: Apartado de comunicaciones del proyecto.	44
Ilustración 42: Creación de dispositivo de E/S.	44
Ilustración 43: Importar tags desde el explorador de proyectos.	45
Ilustración 44: Importación de tags usando OPC.	45
Ilustración 45: Exportación de variables en Unity Pro XL.	46
Ilustración 46: Apartado comunicaciones en el explorador de proyectos.	47
Ilustración 47: Creación de nuevo servidor.	47
Ilustración 48: Selección de tipo del dispositivo de E/S.	47
Ilustración 49: Selección de Modbus como método de comunicación.	48
Ilustración 50: Selección de dirección IP y puerto.	48
Ilustración 51: Posibilidad de vincular con una base externa de datos.	48
Ilustración 52: Apartado Comunicaciones en el explorador de proyectos.	49
Ilustración 53 Modificación de la configuración del dispositivo de E/S.	49
Ilustración 54: Importación de tags desde el explorador de proyectos.	50
Ilustración 55: Ventana de exportación de variables.	50
Ilustración 56: Direccionado del vector y su cabecera en Unity Pro XL.	51
Ilustración 57: configuración del vector en Vijeo Citect.	51
Ilustración 58: Esquema general de la instalación con sensores y actuadores del sistema de transporte.	54
Ilustración 59: Sensores inductivos ubicados en el final del tramo.	55
Ilustración 60: Retenedores (actuadores) del sistema de transporte.	56
Ilustración 61: Panel de control de la instalación.	57
Ilustración 62: Guía GEMMA [14].	58
Ilustración 63: Guía GEMMA implementada en la instalación.	59
Ilustración 64: Configuración de los módulos disponibles en el autómata.	61
Ilustración 65: Acceso a la ventana de conexión al autómata.	61
Ilustración 66: Cuadro para establecer conexión con el autómata usando la red Ethernet.	62
Ilustración 67: Ventana emergente para configurar la dirección.	62
Ilustración 68: Esquema general de la distribución de sensores y actuadores.	69
Ilustración 69: Panel de control del alimentador de piezas.	71

Ilustración 70: Ventana de configuración de Unity Pro XL.	74
Ilustración 71: Pantalla Home con distintos errores en los equipos.	80
Ilustración 72: Cuadro de mandos en espera de puesta en marcha y en espera de reseteo de error grave.	81
Ilustración 73: Pantalla <i>Info Instalación</i> mostrando una parada de emergencia.	81
Ilustración 74: Ventana de transporte mostrando estimaciones de las posiciones de las bandejas.	82
Ilustración 75: Pantalla <i>Almacén Matricial</i> mostrando un error de comunicación.	83
Ilustración 76: Ventana del alimentador de bandeja sin presencia de fallos.	83
Ilustración 77: Alimentador de bandejas en caso de fallo grave.	84
Ilustración 78: Ventana de ayuda de la aplicación SCADA.	84
Ilustración 79: Pantalla de visualización de alarmas.	86
Ilustración 80: Registro de la actividad en un archivo de texto.	87

Lista de abreviaturas

<i>ASRS</i>	Automated Storage and Retrieval System; Sistema de almacenamiento y extracción automatizado.
<i>OPC</i>	OLE for Process Control; OLE para control de procesos.
<i>OLE</i>	Object Link and Embedding; Enlace e incrustación de objetos.
<i>SCADA</i>	Supervisory Control and Data Acquisition; Supervisión, control y adquisición de datos.
<i>ACL</i>	Advanced Control Language; Lenguaje de control avanzado.
<i>SCARA</i>	Selective Compliant Articulated Robot Arm; Brazo robótico articulado de respuesta selectiva.
<i>PLC</i>	Programmable Logic Controller; Controlador lógico programable.
<i>DCD</i>	Data Carrier Detect
<i>RXD</i>	Receive Data
<i>TXD</i>	Transmit Data
<i>DTR</i>	Data Terminal Ready
<i>GND</i>	Signal Ground
<i>DSR</i>	Data Set Ready
<i>RTS</i>	Request to Send
<i>CTS</i>	Clear to Send
<i>RI</i>	Ring Indicator
<i>CRC</i>	Cyclic Redundancy Check; Códigos de redundancia cíclica.
<i>GEMMA</i>	Guide d'Etudes des Modes de Marches et d'Arrêts; Guía de estudio de los modos de marcha y paro.
<i>ADEPA</i>	Agence Nationale pour le Developpement de la Production Appliquée a l'industrie; Agencia nacional francesa para el desarrollo de la producción aplicada a la industria.
<i>AFNOR</i>	Association Française de Normalisation; Agencia francesa de normalización.

1 INTRODUCCIÓN

En el siguiente documento se encuentra una descripción de la integración de distintos equipos en una célula de fabricación flexible.

En primer lugar, se presentarán los objetivos del proyecto y una descripción de la célula de fabricación flexible situada en los laboratorios de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla.

Tras la descripción, se presentarán las distintas posibilidades en las conexiones entre los equipos. Se hará distinción entre la conexión entre autómatas, y la conexión entre el autómatas y el controlador del sistema robotizado de almacenaje.

También se presentarán dos métodos de conexionado de los autómatas al SCADA, desarrollado en Vijeo Citect. Estos métodos de conexión serán comparados entre ellos en una tabla resumen.

Por último, se explicará la aplicación práctica desarrollada, presentando la programación de cada una de las máquinas que componen la célula de fabricación flexible, así como la aplicación SCADA realizada para la supervisión de la instalación.

2 OBJETIVOS

Este trabajo se encuadra dentro de los denominados proyectos de desarrollo, donde se hacen uso de distintas herramientas y equipos comerciales de automatización.

Se utiliza como planta de automatización una célula de fabricación flexible instalada en el departamento de Ingeniería de Sistemas y Automática (Escuela Técnica Superior de Ingeniería, Universidad de Sevilla). Esta planta es utilizada como plataforma de formación del alumnado y lleva funcionando varios años, aunque durante la realización de este trabajo se estaban llevando a cabo varias modificaciones y ampliaciones de la misma.

El trabajo se centra en abordar el problema que surge en una célula de fabricación donde coexisten equipos de cierta antigüedad con equipos más modernos y de distintos fabricantes, pero que deben funcionar de forma coordinada para llegar a objetivo de producción.

Concretando las líneas de trabajo se focalizan en los siguientes puntos:

- 1- Elección y programación de un equipo principal que funcione como maestro en relación al resto de equipos de la instalación y maneje el sistema de transporte. Este equipo contemplará la posibilidad de comunicación avanzada (modbus usando Ethernet y comunicación puerto serie).
- 2- Integración de un equipo de almacén robotizado de pallet (Automated Storage and Retrieval System, ASRS), más antiguo y que admite comunicación serie.
- 3- Integración de un sistema de almacenamiento de bandejas, basado en equipo con posibilidades de comunicación avanzada (modbus/Ethernet).
- 4- Supervisión mediante SCADA (sistema de supervisión, control y adquisición de datos), explorando distintas opciones de conexionado y comunicación.

Se hará especial hincapié en la comunicación entre equipos planteando distintas opciones. En estas comunicaciones serán tratados aspectos de conexión física, configuración de equipos y desarrollo de programas.

Concretamente, se desarrollarán las siguientes comunicaciones:

- a) Conexión entre autómatas. Equipos M340 de Schneider Electric basado en Modbus sobre Ethernet.
- b) Conexión del autómata (M340) con el controlador del ASRS de Eshed Robotic, basado en envío de mensajes de caracteres por puerto serie. Se tendrá que definir un protocolo de comunicación y proponer un algoritmo CRC (Código de Redundancia Cíclica) con el fin de poder detectar fallos en el envío de los mensajes.
- c) Comunicación entre el SCADA (Vijeo Citect) y el PLC M340, o la comunicación simultánea entre el Vijeo Citect y los dos autómatas M340.

La comunicación del SCADA con los distintos autómatas se llevará a cabo mediante OPC (OLE¹ for Process

¹ Object Linking and Embedding.

Control), comunicación general, o mediante conexión Unity Speedlink, comunicación propietaria entre equipos de Schneider Electric, planteando las ventajas e inconvenientes de cada forma de enlace.

Por último y a modo de aplicación se implementará un ejemplo tipo “demo” donde se puede comprobar los programas desarrollados.

3 DESCRIPCIÓN DE LA INSTALACIÓN

En este capítulo se llevará a cabo la descripción de la instalación, la cual ha sido una célula de fabricación flexible ubicada en la planta baja de los laboratorios del *Departamento de Ingeniería de Sistemas y Automática* de la *Escuela Técnica Superior de Ingeniería*.

3.1. Células de fabricación flexible

En primer lugar, se hará una breve introducción sobre las células de fabricación flexible.

Una célula de fabricación flexible consiste en un conjunto de estaciones de trabajo interconectadas entre ellas por un sistema de transporte automatizado, cuya programación se realiza de forma modular (su programación no se debe ver caracterizada como un ciclo repetitivo y cerrado, sino como un ciclo flexible, adaptable los distintos objetivos de producción). La gestión coordinada de estos módulos de programación permite alcanzar a la instalación los objetivos marcados en ese momento.

El campo de aplicación de la fabricación flexible cubre el vacío entre la producción con automatización fija (caracterizada por una secuencia de procesamiento fija y grandes cantidades de producción) y la producción con automatización programable (caracterizada por tasas de producción bajas y producción en lotes).

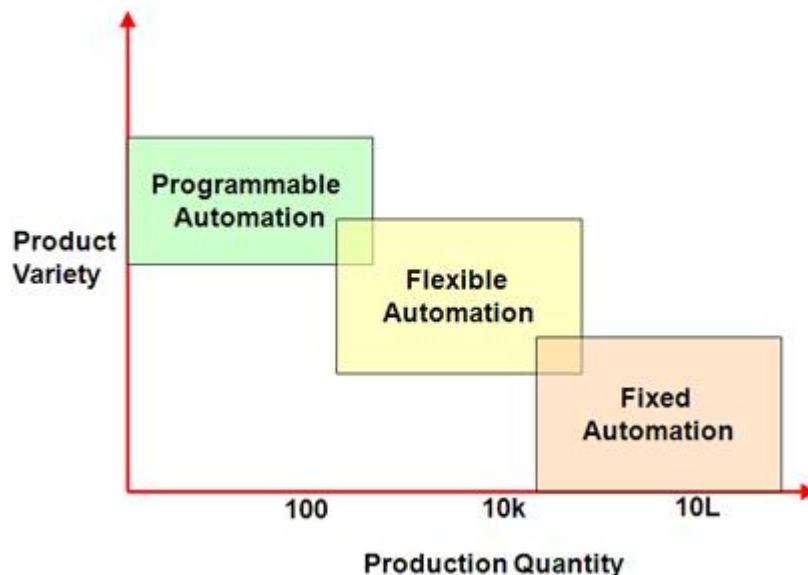


Ilustración 1: Diferencias entre la variación del producto y de la cantidad para los distintos tipos de automatización [1].

3.2. Descripción de la instalación utilizada

En este apartado se describirán los equipos disponibles en la célula de fabricación flexible ubicada en los

laboratorios del Departamento de Ingeniería de Sistemas y Automática.

La instalación se compone de distintos puestos de trabajo, los cuales se encuentran conectados por medio de un circuito de cintas transportadoras. Sobre estas cintas transportadoras se colocarán bandejas, sobre las cuales se pondrán los pallets que serán usados para el transporte de piezas.

En los siguientes sub-apartados que se encuentran a continuación se presentan todas las máquinas disponibles en la instalación.

3.2.1 Circuito de cintas transportadoras

Consiste en cinco cintas transportadoras, en las que las cuatro primeras se encuentran formando un circuito rectangular, mientras que la quinta se encuentra de forma perpendicular a la cinta 1, en forma de T. El sentido de la marcha en cada cinta es único, excepto en el caso de la cinta 5, en la que es posible el movimiento en ambas direcciones, así como una regulación en su velocidad.

Este circuito se encuentra controlado por un PLC de Schneider Electric modelo M340, el cual se considerará el PLC principal de toda la instalación.

El circuito de transporte cuenta con 14 retenedores de tipo neumático (marcados en el esquema como Y) y 19 sensores de tipo inductivo (marcados en el esquema como S).

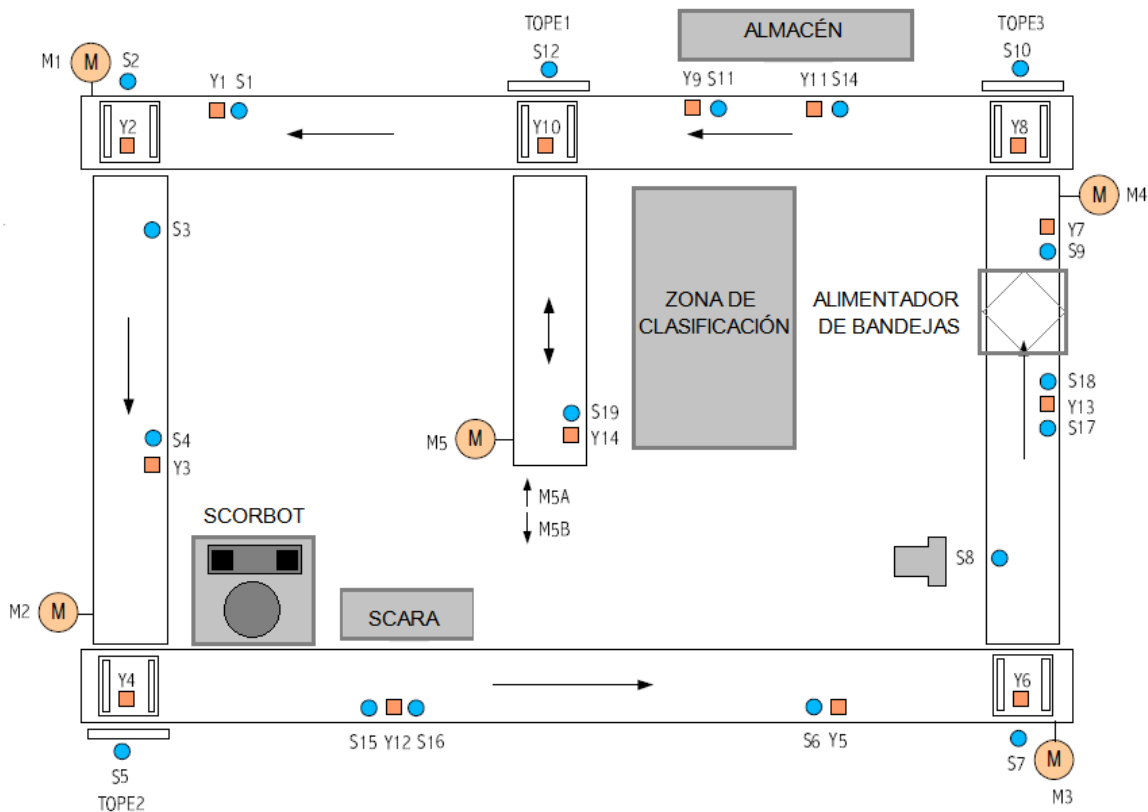


Ilustración 2: Esquema general de la planta.

3.2.2 Almacén matricial ASRS de Eshed Robotic

Es un almacén robótico matricial cuya función principal será servir pallets a aquellas bandejas que los necesiten o almacenar los pallets colocados encima de las bandejas.

Dispone de 72 compartimentos divididos en 2 estanterías, los cuales el robot manipula desde el hueco entre ellos. Posee 4 grados de libertad (ejes X, Y, Z y un ángulo de giro theta) así como una pinza neumática como

actuador. Cuenta también con un sensor que fue instalado para permitir al almacén distinguir entre estantes vacíos y llenos.

El almacén es controlado por un Controlador-B usando el lenguaje ACL (Advanced Control Language, lenguaje de control avanzado). Permite tanto control de trayectoria como de punto a punto, y multitarea. El controlador además dispone de conexión por puerto serie con el autómatas principal (la conexión se encuentra detallada en la sección de comunicación del autómatas con el almacén).



Ilustración 3: Almacén matricial ASRS.

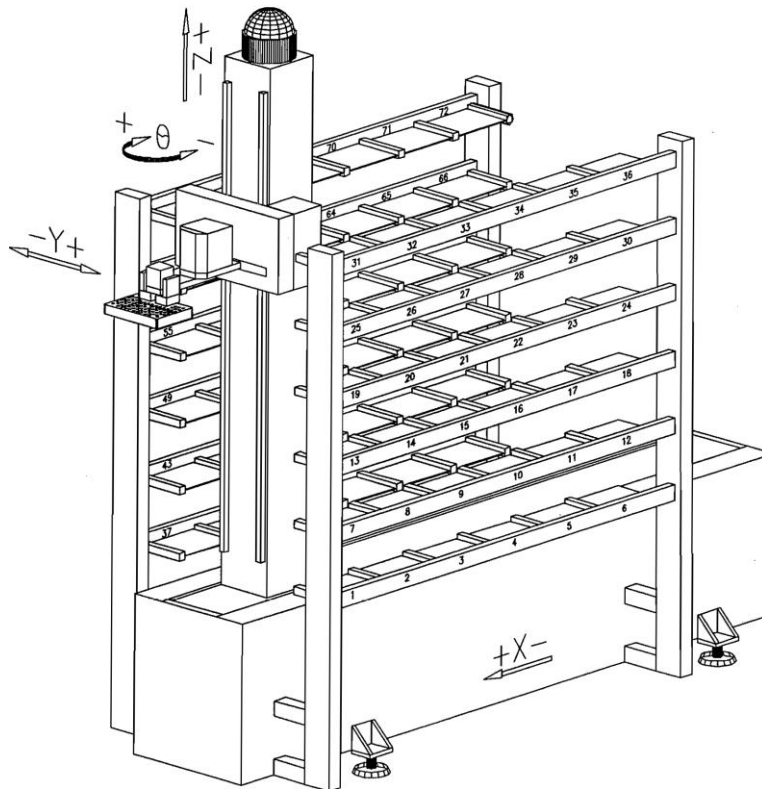


Ilustración 4: Almacén matricial de la célula de fabricación flexible [2].

3.2.3 SCORBOT-ER 4u

Consiste en un robot con 5 ejes de libertad con el que se puede manipular el contenido de los pallets. Cuenta con conexión a un autómatas del mismo tipo que el principal situado en la parte inferior del puesto de trabajo. Este autómatas se encuentra también conectado al PLC principal mediante conexión Ethernet.

Se encuentra ubicado en la intersección entre los tramos 2 y 3. Actualmente se está instalando un dispensador de piezas realizado con un motor paso a paso que permita usar el scorbot para almacenar o servir piezas a los pallets. En la siguiente ilustración se muestra el dispensador de bandejas y el scorbot en la instalación marcados como números 1 y 2 respectivamente.

3.2.4 Robot Sony SCARA

Robot tipo SCARA (Selective Compliant Articulated Robot Arm, brazo robótico articulado de respuesta selectiva) situado en el tramo 3. Se puede apreciar en la Ilustración 5: Puesto de trabajo del Scorbot con el dispensador de piezas y robot SCARA. como número 3. Actualmente se encuentra fuera de servicio.

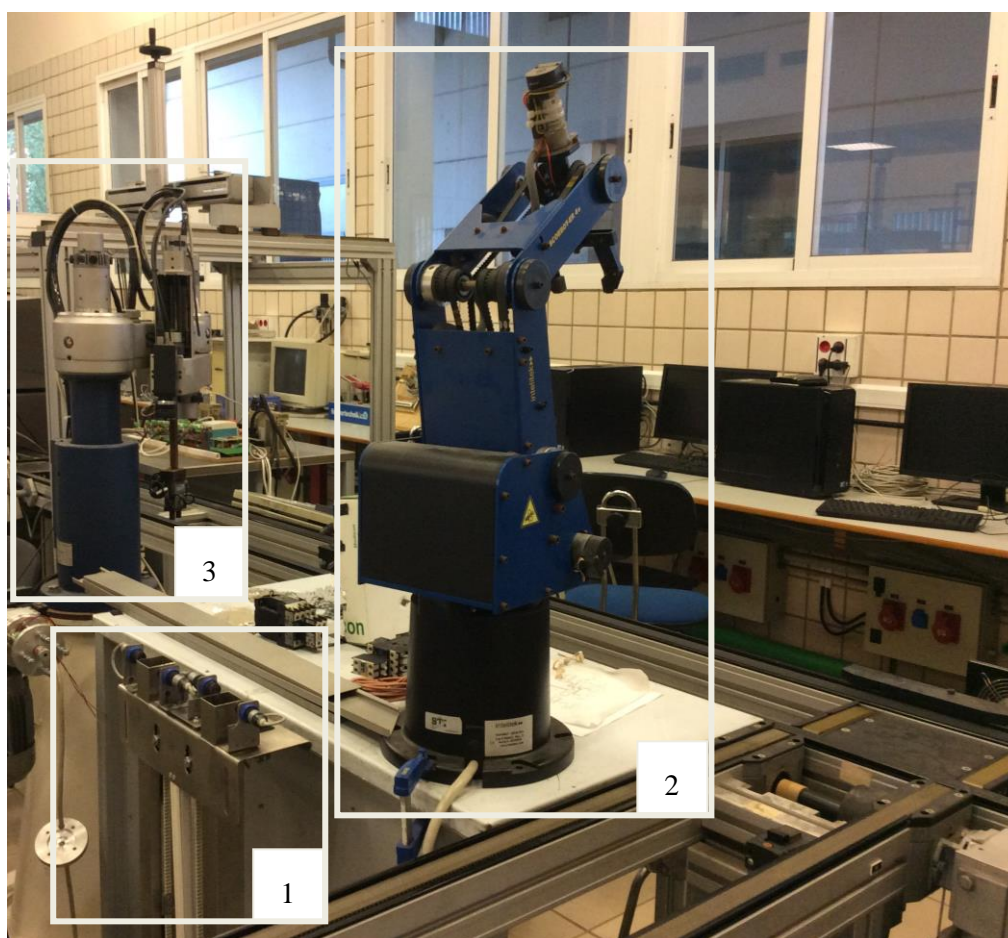


Ilustración 5: Puesto de trabajo del Scorbot con el dispensador de piezas y robot SCARA.

3.2.5 Alimentador de bandejas

Se encuentra ubicado en el tramo 4 del sistema de transporte. Su función es almacenar o servir bandejas al sistema de transporte. Cuenta con un PLC Schneider Electric M340 conectado por Ethernet al PLC principal ubicado en el cuadro de mandos central.



Ilustración 6: Dispensador de bandejas.

3.2.6 Zona de clasificación

Actualmente se encuentra en proceso de instalación. Se compone de los siguientes elementos:

- Alimentador de piezas: Permite sacar o almacenar piezas. Usa un motor paso a paso y su control se realiza mediante el PLC principal.
- Pórtico: Permite desplazar piezas entre el alimentador de bandejas, la cinta transportadora de piezas o el tramo 5 de las cintas. Utiliza un motor paso a paso.
- Cinta transportadora de piezas: Comunica la zona del pórtico con la mesa giratoria de clasificación. Utiliza para ello un motor paso a paso.
- Mesa de clasificación: Mesa circular con una serie de sensores que permiten la clasificación de la pieza. Permite el reconocimiento del color, la medida de la pieza y el material de la pieza. El giro se controla mediante un motor paso a paso.

4 COMUNICACIÓN ENTRE EL ALMACÉN ROBOTIZADO DE PALLETS Y EL PLC PRINCIPAL

El equipo robotizado de almacenamiento de pallet que se encuentra en la instalación es un ASRS de Eshed Robotic, un equipo de cierta antigüedad que se pretende integrar en la instalación de forma que funcione de forma coordinada con el resto de los equipos.

El controlador del equipo admite una conexión serie que permitirá la integración en la instalación. Para la conexión se han consultado el manual de prácticas del Laboratorio de Informática y Robótica Industrial [3] y el manual de usuario del Modicon M340 [4].

En primer lugar, se hará una descripción de la conexión hardware de los equipos, luego una descripción de la configuración necesaria, y por último el desarrollo de los bloques de programación básicos para ello.

4.1. Conexión del hardware

4.1.1 Conexión física

El almacén robotizado es controlado por un Controlador-B de Eshed Robotic, que dispone de 2 puertos RS232 (para consola y COM1) con conector DB9 macho. Estos puertos son ampliables hasta 10 (en este caso de tipo D37 hembra).

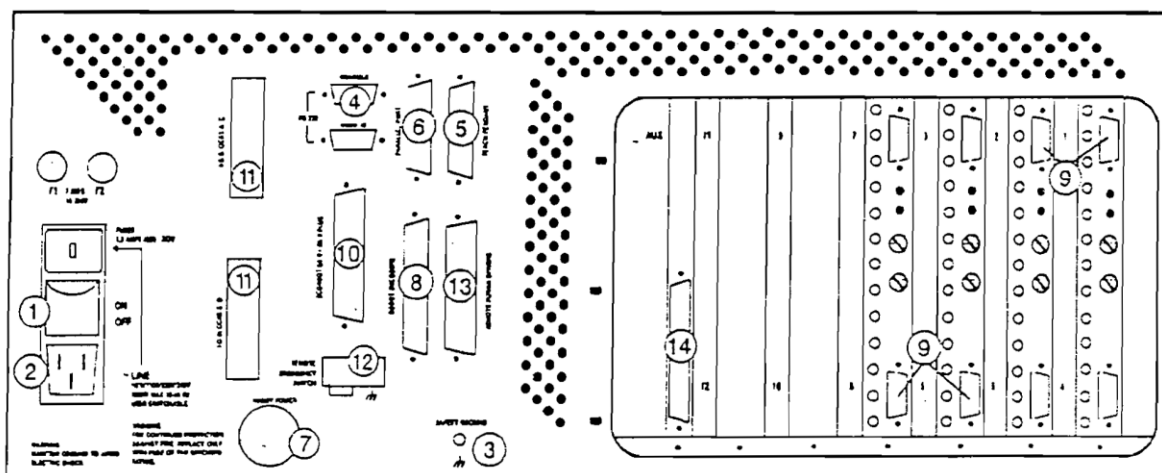


Ilustración 7: Panel trasero del Controlador-B [5].

Tabla 1: Panel trasero del Controlador-B.

1	Interruptor de encendido/apagado.		8	Conector de encoders del robot	D37 macho
2	Conector de entrada tensión de red		9	Conectores de los drivers de los ejes	D9 hembra
3	Terminal de tierra de seguridad		10	Conector ER Vplus; reservado	D50; NO USADO
4	Canales RS232: Consola y COM1	D9 macho	11	Conectores entrada/salida	Centronic 36-pin
5	Conector botonera de enseñanza	D25 macho	12	Interruptor emergencia remoto	
6	Conector puerto paralelo	D25 macho	13	Remote servo driver command output conn.	D50 macho
7	Conector de potencia del robot	19-pin round hembra	14	Multipuertos RS232 (opcional)	D37 hembra

El autómatas principal utilizado es un Schneider Electric Modicon M340 con un módulo procesador BMX P34 2020, que dispone de conexión USB, Ethernet y puerto serie (admitiendo comunicación en modalidad MODBUS y de caracteres). En la siguiente imagen se ilustra la descripción física del mismo.

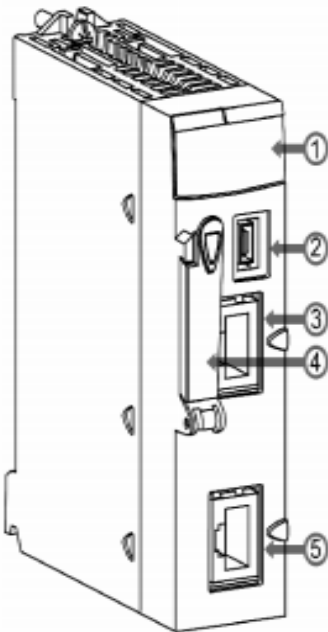


Tabla 2: Descripción de los elementos del módulo.

1	Pantalla de indicadores LED
2	Puerto USB
3	Puerto Ethernet
4	Ranura de la tarjeta de memoria SD
5	Puerto serie MODBUS (conector RJ45)

Ilustración 8: Descripción física del módulo [6].

La comunicación hardware entre equipos se ha realizado mediante un cable con conector DB9 hembra (que irá conectado en el extremo del controlador del almacén matricial) y conector RJ45 en el extremo que va conectado al PLC. Esta conexión permite el intercambio de datos mediante el modo de caracteres.

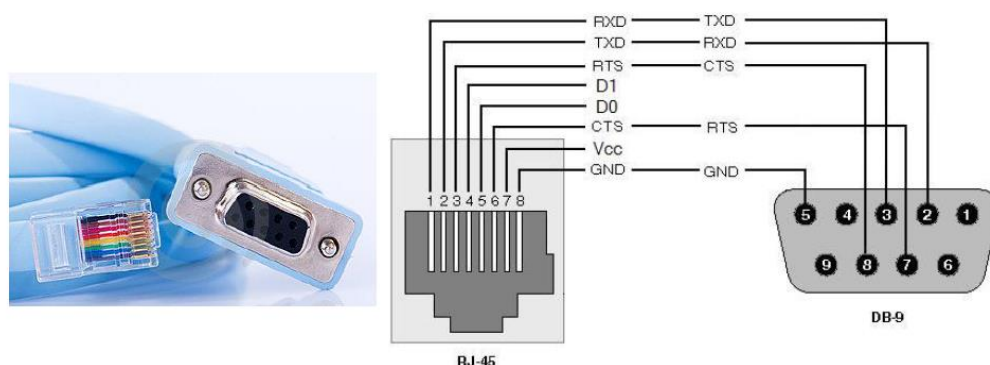


Ilustración 9: Correspondencia entre pines de los conectores [3].

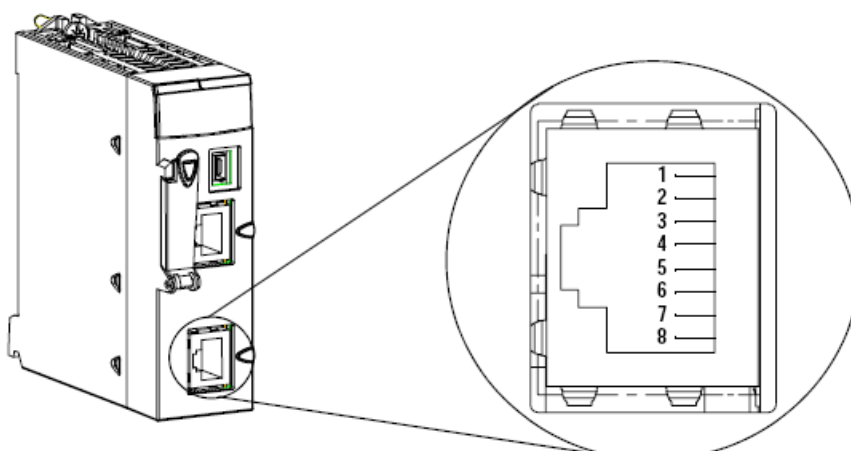


Ilustración 10: Detalle del puerto RJ45 de los procesadores BMX P34 1000/2010/2020 [4].

En la Tabla 3 se especifica el uso de cada pin en el conector RJ45, mientras que en la Tabla 4: detalles de los pines del conector DB9 se especifica el uso de los pines del conector DB9.

4.1.2 Configuración del hardware

Con el fin de establecer la conexión entre el PLC y el autómat, será necesario configurar los parámetros de velocidad y la existencia de bits de paridad o los bits de parada.

En el caso del Controlador-B de Eshed Robotic, en el manual de usuario [5] se especifica que la comunicación se hace en canales RS232 a 9.600 bauds, con 8 bits de datos, sin paridad y con protocolo XON/XOFF (sin posibilidad de cambio de la configuración), mientras que la comunicación en el módulo procesador del autómat en su modo de caracteres permite la configuración de diversos parámetros como los bits de paridad, la velocidad de transmisión o el tipo de línea física. Por ello, el equipo que deberá ser configurado en este caso es el autómat.

Para ajustar los parámetros del autómat, en el explorador de proyectos se pulsará en el desplegable de la tarjeta BMX P34 2020; en la imagen del módulo procesador se pulsará en el puerto *SerialPort*, dando acceso a la pantalla mostrada en la Ilustración 11. En ella se pueden modificar la velocidad de transmisión, parámetros de detección de final de mensaje, y parámetros de la línea física y señales.

Tabla 3: Detalle de los pines del conector RJ45.

1	RXD
2	TXD
3	RTS
4	D1
5	D0
6	CTS
7	Fuente de alimentación
8	Común

Tabla 4: detalles de los pines del conector DB9.

1	DCD: Data Carrier Detect
2	RXD: Receive Data
3	TXD: Transmit Data
4	DTR: Data Terminal Ready
5	GND: Signal Ground
6	DSR: Data Set Ready
7	RTS: Request to Send
8	CTS: Clear to Send
9	RI: Ring Indicator

Los parámetros correctos para la conexión en modalidad de caracteres son los mostrados en la Tabla 5. Además de estos parámetros, también se debe indicar el carácter usado para la detección del final del mensaje. En este caso, se detectará el final de mensaje mediante el uso del carácter especial '/', equivalente al código ASCII 47.

Tabla 5: Parámetros para la configuración del modo de caracteres.

Parámetros	Valor
Línea física	RS 232
Velocidad de transmisión	9600 bits/s
Paridad	Ninguna
Bits de datos	8 bits
Bits de parada	1 bit
Señales	RX/TX

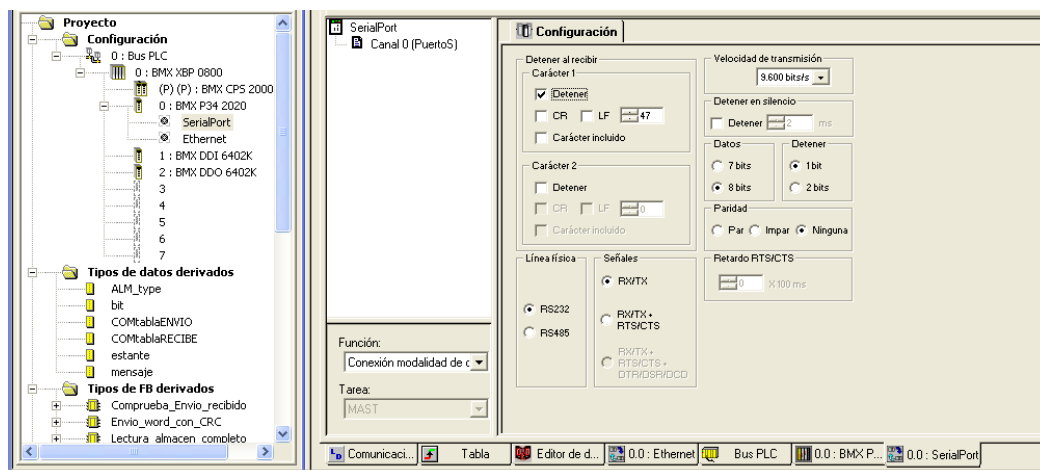


Ilustración 11: Pantalla de configuración de la conexión en modo de caracteres.

4.2. Comunicación maestro-esclavo

La comunicación entre los dos dispositivos se ha planteado como una comunicación asíncrona maestro-esclavo, donde el PLC principal será el que dirige la comunicación. El maestro es el que inicia la comunicación, mientras que el esclavo se encarga de responder a las peticiones recibidas desde el PLC. El intercambio de información se hace mediante una cadena de diez bits en los que los seis primeros constituyen el mensaje, mientras que los cuatro últimos son el resultado de un algoritmo de redundancia.

La comunicación se realiza en la modalidad de caracteres, por lo que es necesaria la conversión de estos mensajes binarios a cadenas, dando lugar a mensajes de 10 bytes (en vez de 10 bits como sería de esperar).

En el diagrama de la Ilustración 12 se han representado las distintas posibilidades que se pueden dar para esta comunicación desde el punto de vista del autómat, que ejercerá el papel de maestro.

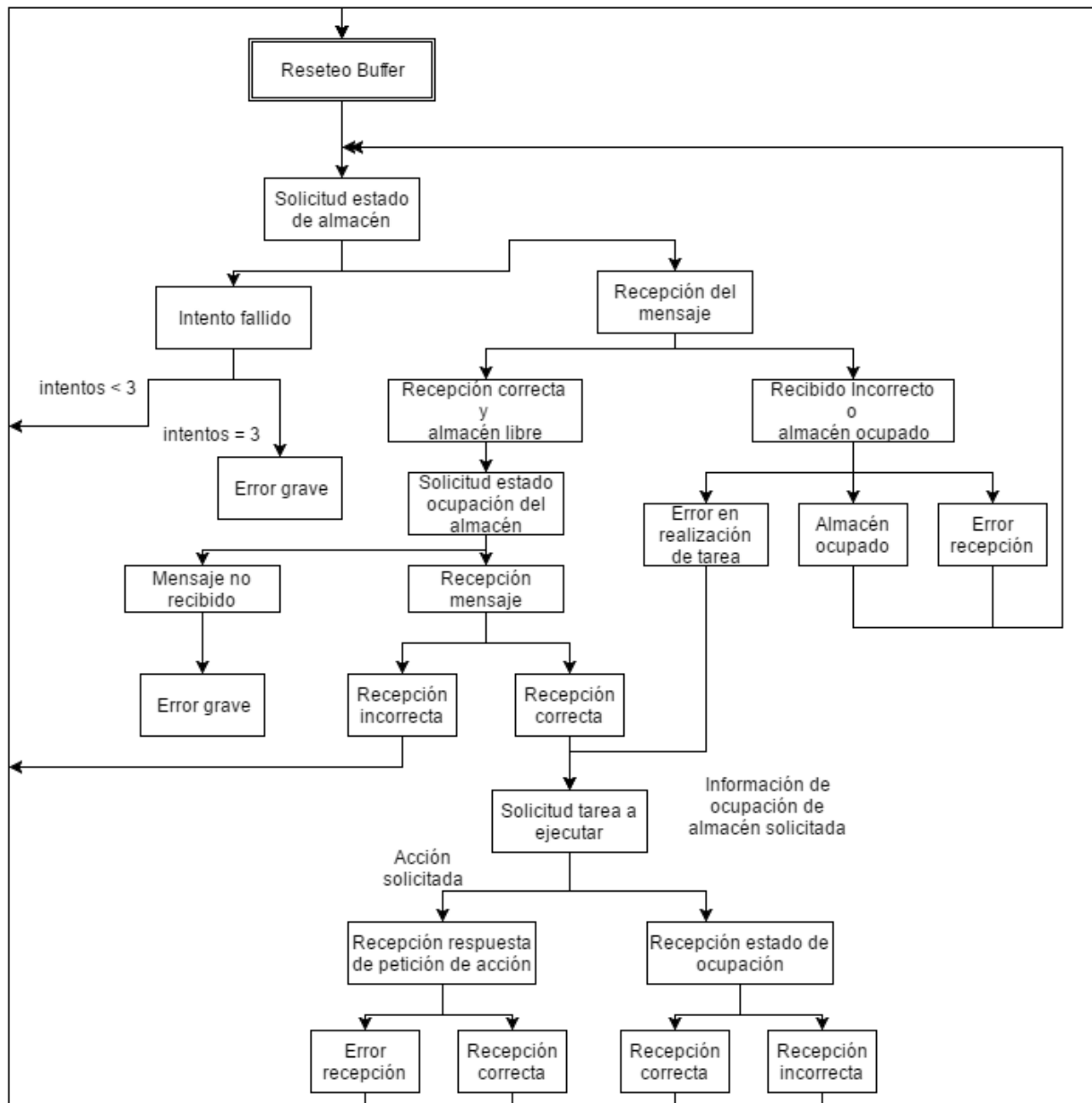


Ilustración 12: Posibilidades en la comunicación.

Es de destacar el caso en el que se dé un error de la realización de la operación, que será considerado como almacén ocupado en la condición de entrada, pasando al error de realización de tarea. También hay que tener en cuenta que, en el caso de error grave, se deberá resetear la comunicación por completo tras haber solucionado el fallo de conexión.

4.2.1 Estructura del mensaje

Los mensajes se componen de 10 caracteres que podrán ser 0 (correspondientes a ASCII 48) o 1 (ASCII 49).

En la Tabla 6 se encuentra la estructura del mensaje usada para las comunicaciones. Como se comentó anteriormente, los 6 primeros bytes (del 0 al 6) corresponden a los caracteres que componen el mensaje, mientras que los siguientes cuatro bytes (del 6 al 9) contendrán los cuatro caracteres que forman el CRC.

Tabla 6: Estructura del mensaje de petición y de respuesta a la petición.

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

4.2.1.1 Envío de peticiones

En la Tabla 7 se presentan los mensajes que el autómata envía para solicitar peticiones al controlador del almacén.

Tabla 7: Mensajes de peticiones.

Mensaje intercambiado	Operación solicitada	Operación detallada
0 0 0 0 0 0	Sacar pallet	Saca pallet procesado
0 0 0 0 0 1	Sacar pallet	Saca pallet sin procesar
0 0 0 0 1 0	Sacar pallet	Saca pallet defectuoso
0 0 1 0 0 0	Meter pallet	Mete pallet procesado
0 0 1 0 0 1	Meter pallet	Mete pallet sin procesar
0 0 1 0 1 0	Meter pallet	Mete pallet defectuoso
0 1 0 0 0 0	Obtener información	Obtener estado actual
0 1 0 0 0 1	Obtener información	Forzar un chequeo del almacén
0 1 0 0 1 0	Obtener información	Enviar matriz de ocupación completa

4.2.1.2 Respuesta de peticiones

En primer lugar, el PLC solicitará el estado actual del almacén, es decir: si el almacén se encuentra llevando a cabo una petición, si la manipulación de pallets en el recurso compartido ha terminado o si se encuentra a la espera de una nueva petición.

- En el caso de que el almacén esté en espera de una nueva petición, responderá '1 0 0 0 0 0'.
- En el caso de que se esté llevando a cabo una operación de servir o extraer pallet en el almacén y ya se haya terminado de trabajar con el recurso compartido (el puesto de trabajo en la cinta donde el almacén recoge o deposita el pallet), se realizará un envío con mensaje '0 0 1 1 1 1' indicando que la bandeja puede continuar moviéndose en la cinta.
- Si el almacén se encuentra ocupado actualmente, entonces se responderá con el mismo mensaje con el que se solicitó la tarea que se está llevando a cabo. Por ejemplo, si el almacén se encuentra realizando un chequeo del almacén, cada vez que se le solicite el estado responderá como '0 1 0 0 0 1', siendo el primer carácter el que indica que se encuentra ocupado, y el resto de caracteres de la petición que está realizando (chequeando el almacén).
- En el caso de que el almacén haya tenido un error durante la ejecución de alguna petición, responderá

con el mensaje correspondiente a la petición que estaba llevando a cabo cuando se produjo el error, aunque con los caracteres 2 y 4 con un '1'.

A continuación en la Tabla 8 se muestran todos los posibles mensajes recibidos por el autómata.

Tabla 8: Mensajes posibles de respuesta del almacén.

Mensaje	Tipo de mensaje	Explicación
1 0 0 0 0 0	Información	Espera nueva petición
0 0 1 1 1 1	Información	Recurso compartido liberado
0 0 0 0 0 0	Información	Sirviendo pallet procesado
0 0 0 0 0 1	Información	Sirviendo pallet sin procesar
0 0 0 0 1 0	Información	Sirviendo pallet defectuoso
0 0 1 0 0 0	Información	Almacenando pallet procesado
0 0 1 0 0 1	Información	Almacenando pallet sin procesar
0 0 1 0 1 0	Información	Almacenando pallet defectuoso
0 1 0 0 0 1	Información	Haciendo un chequeo físico de la ocupación del almacén
0 1 0 1 0 0	Error	Error al servir pieza procesada: no hay pieza de ese tipo en el almacén.
0 1 0 1 0 1	Error	Error al servir pieza sin procesar: no hay pieza de ese tipo en el almacén.
0 1 0 1 1 0	Error	Error al servir pieza defectuosa: no hay pieza de ese tipo en el almacén.
0 1 1 1 0 0	Error	Error al almacenar pieza procesada: no hay sitio para ese tipo de pieza en el almacén.
0 1 1 1 0 1	Error	Error al meter almacenar pieza sin procesar: no hay sitio para ese tipo de pieza en el almacén.
0 1 1 1 1 0	Error	Error al almacenar pieza defectuosa: no hay sitio para ese tipo de pieza en el almacén.

4.3. Algoritmos de redundancia

Se presenta una breve introducción teórica sobre los CRC extraída del artículo [7].

Los algoritmos de redundancia permiten proteger los datos de errores añadiendo una redundancia de información. En un caso general, los datos a ser protegidos se dividen en bloques de k-bits. A partir de cada bloque anterior se genera un bloque de n bits, obteniendo n-k bits de información redundante. Estos n-k bits se conocen como bits de chequeo o de paridad.

La selección del número de bits de información redundante para una aplicación específica depende de factores en los que se incluyen la protección necesaria, el coste de la implementación, la estrategia del control del error o la naturaleza de los errores. En general, a mayor número de bits de chequeo, mayor poder de detección de

errores.

Antes de tratar los códigos de redundancia cíclicos, se presentará el concepto de campo binario (y las operaciones en él) y de polinomios binarios, necesarios para una mejor comprensión.

Un campo es un sistema algebraico en el que se pueden realizar las operaciones de suma, resta, multiplicación y división. El conjunto de números reales, por ejemplo, constituye un campo. Éstos pueden ser finitos o infinitos.

El campo matemático finito más reducido posible es el binario, constituido por sólo dos elementos denotados como 0 y 1. A continuación, se muestran en la tabla las operaciones de suma y de multiplicación en este campo.

Tabla 9: Operación de suma en el campo algebraico binario.

+	0	1
0	0	1
1	1	0

Tabla 10: Operación de multiplicación en el campo algebraico binario.

·	0	1
0	0	0
1	0	1

Se aprecia como la suma es lo mismo que la resta en este campo, así como que la multiplicación puede ser implementada con una puerta AND. Además, en el caso de la división sólo se definirá en este campo entre el elemento 1, dejando el valor 1 o 0 inalterado.

Un polinomio binario es un polinomio cuyos coeficientes han sido tomados del campo binario. Dada una secuencia de bits, es posible asociarle un polinomio binario viendo los distintos bits como una representación de los coeficientes del polinomio. Por ejemplo, la secuencia 101011 se puede asociar al polinomio de quinto grado $1 \cdot x^0 + 0 \cdot x^1 + 1 \cdot x^2 + 0 \cdot x^3 + 1 \cdot x^4 + 1 \cdot x^5 = 1 + x^2 + x^4 + x^5$.

Los códigos de redundancia cíclicos (también conocidos como códigos polinomiales o CRC) (Cyclic Redundancy codes) preservan la integridad de los datos en aplicaciones de transmisión de datos (son especialmente adecuados para la detección de errores de ráfaga ²). En ellos, se eligen los bits de chequeo de forma que los polinomios asociados son múltiplos de un determinado polinomio denominado polinomio generador.

A continuación, se expone un procedimiento de cálculo de CRC dado su polinomio generador $g(x)$ de grado $n-k$:

$u(x)$ y $v(x)$ representan los polinomios asociados a un mensaje de k bits (mensaje polinomial) y su correspondiente bloque generado de n bits respectivamente. La relación entre $u(x)$ y $v(x)$ puede entonces expresarse como

$$v(x) = u(x) g(x)$$

Nótese que el número de bits con información redundante es igual al grado del polinomio generador $g(x)$ (en este caso, $n-k$). Además, este procedimiento sólo implica la división de dos números binarios, por lo que puede ser fácilmente implementado con registros de desplazamientos y puertas XOR.

4.3.1 Cálculo de CRC

El cálculo del CRC realizada en los equipos consiste en hacer una división del mensaje binario que se quiere enviar entre el polinomio del CRC escogido, de forma que el resto de esta división entre polinomios dará lugar al valor del CRC del mensaje (nuestro objetivo).

Los pasos seguidos para el cálculo del CRC son:

1. Elección de un polinomio por el cual será dividido el mensaje. En este caso se ha elegido el polinomio

² Errores ocasionales en la transmisión de datos que afectan a varios bits seguidos.

10011.

2. Creación del mensaje ampliado: Antes de calcular el CRC, se añaden al mensaje tantos 0 como la posición del bit más significativo en el polinomio elegido. En este caso, la posición del bit más significativo es la cuarta, por lo que se han añadido 4 bits con valor 0.
3. División del mensaje ampliado por el polinomio elegido, siendo el resto el valor del CRC buscado.

4.3.2 Implementación de la división de polinomios

Para la implementación de la división se ha optado por la más sencilla posible, debido a las propias limitaciones del controlador del almacén (ausencia de puertas OR exclusivas, ausencia de registros de desplazamientos, etc.). Además de esto, en la comunicación que se lleva a cabo el tiempo de ejecución no es un factor limitante, por lo que no se considera de provecho la implementación de un algoritmo mediante consulta de tablas como los que se proponen en [7] o [8].

La implementación se puede llevar a cabo con registros de desplazamientos cuya longitud será igual a la posición del bit más significativo en el polinomio generador elegido; en este caso sólo se necesita un registro de 4 bits.

En el siguiente pseudocódigo extraído de [8] se describen las acciones seguidas para llevar a cabo la división:

```

Cargar el registro con bits a 0.

Crear el mensaje ampliado añadiendo ceros al final del mensaje.

Mientras quede algo en el registro:

    Desplazar el registro hacia la izquierda 1 bit, siendo el bit de
    entrada perteneciente al mensaje ampliado

    Si el bit de salida del registro es igual a 1:

        Registro=Registro XOR Polinomio CRC

    Final del si

Fin mientras

CRC=Registro
  
```

En el caso del controlador del sistema robótico de almacenaje, no existen tipos de variables específicos para la operación con binarios: sólo existen variables enteras de 32 bits y variables tipo STRING, así como vectores de variables enteras. No obstante, sí existen los operadores binarios AND y OR para su uso con enteros. Por ello, todas las operaciones binarias implementadas en el Controlador-B se han llevado a cabo usando vectores como registros de desplazamientos.

En el caso de la implementación en el autómatas, en Unity Pro XL se disponen de diversos tipos de datos elementales (BOOL, EBOOL, INT, DINT, WORD...) así como tipos de datos derivados (ARRAY, STRUCT, ...) disponibles, por lo que los registros de desplazamientos se implementan con el uso de un WORD junto con los bloques de librería SHIFT.

Programación de la comunicación con modalidad de caracteres en UNITY PRO XL

4.3.3 Creación de variable asociada al Puerto serie

Con el fin de realizar la programación de la comunicación, se creará una variable asociada al canal del puerto serie. Para ello, en el explorador de proyectos situado en la zona izquierda superior se pinchará en SerialPort, y

luego en la pestaña Objetos de E/S. En la pantalla mostrada, se busca la dirección asociada al canal (se puede facilitar la búsqueda de la dirección activando la casilla de canal %CH, y pulsando en *Actualizar cuadrícula*). Tras esto, se creará una variable con nombre *PuertoS* de tipo *T_COM_CHAR_BMX* y con la dirección correspondiente del canal.

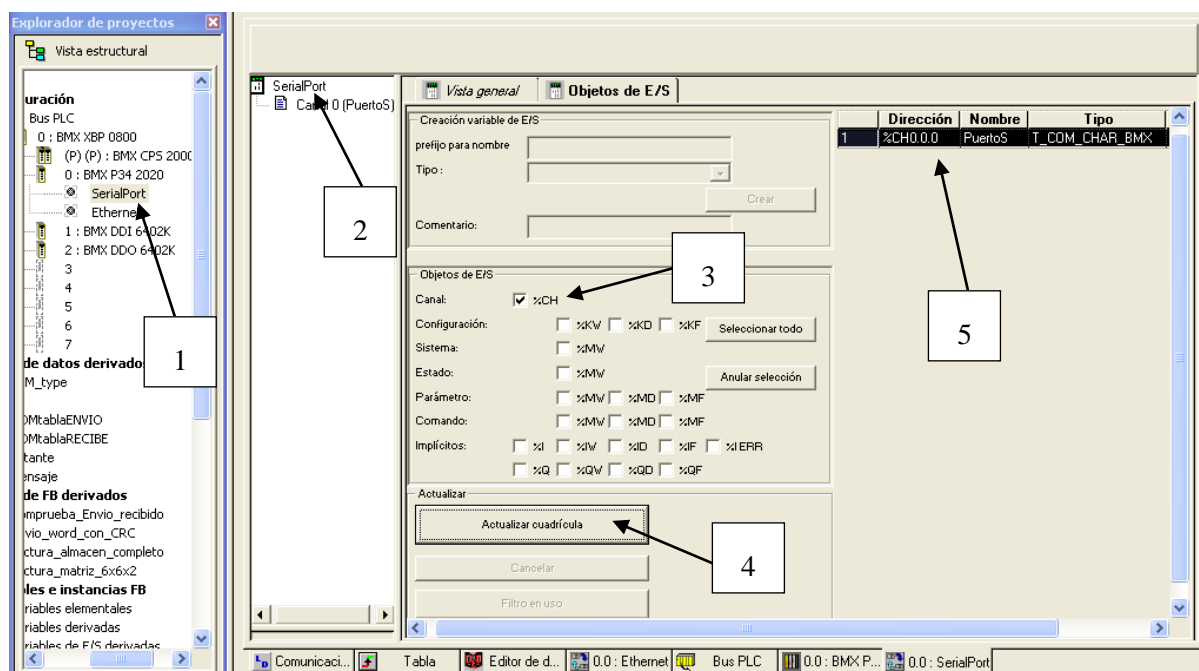


Ilustración 13: Pasos para la creación de la variable asociada al canal.

4.3.4 Librería de bajo nivel usada

Existen tres funciones específicas para enviar y recibir datos mediante la comunicación en modo de caracteres: ADDM, INPUT_CHAR y PRINT_CHAR.

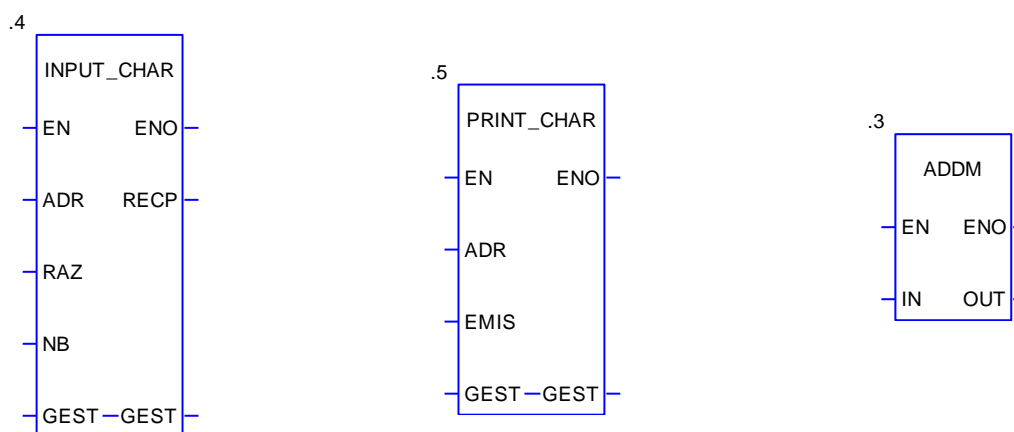


Ilustración 14: Librería de bajo nivel usada en Unity Pro XL.

4.3.4.1 ADDM

Su objetivo es convertir una dirección en cadena de caracteres tipo *STRING* en una tabla de tipo *ARRAY[0...7] OF INT* que pueda ser utilizada por las funciones de comunicación. La entrada siempre será '0.0.0' en este caso.

4.3.4.2 PRINT_CHAR

Permite enviar una cadena de caracteres hasta una longitud máxima de 1.024 bytes. Las entradas y salidas las

se encuentran descritas en la Tabla 11.

Tabla 11: Descripción de los parámetros de PRINT_CHAR.

Parámetros	Tipo	Descripción
ADR	ARRAY[0...7] OF INT	Dirección del destino proporcionada por la función ADDM.
EMIS	STRING	Cadena de caracteres a enviar.
GEST	ARRAY[0...3] OF INT	Tabla de gestión de intercambios.

4.3.4.3 INPUT_CHAR

Permite recibir una cadena de caracteres hasta una longitud máxima de 1.024 bytes. Las entradas y salidas se encuentran descritas en la Tabla 12.

Tabla 12: Descripción de los parámetros de INPUT_CHAR.

Parámetros	Tipo	Descripción
ADR	ARRAY[0...7] OF INT	Dirección del destino proporcionada por la función ADDM.
Reset	INT	Si el valor es 1 resetea el buffer.
NB	INT	Longitud de la cadena de caracteres a leer.
RECP	STRING	Cadena de caracteres recibida.
GEST	ARRAY[0...3] OF INT	Tabla de gestión de intercambios.

4.3.5 Creación de tipos para las tablas de gestión

Con el fin de gestionar la recepción y envío de mensajes, se han creado dos tipos de datos derivados. No obstante, primero será necesario cambiar la configuración del proyecto para que permita matrices dinámicas. Esto puede hacerse accediendo a los *Ajustes del proyecto*, en la pestaña superior *Herramientas*. Una vez allí, se marcará la casilla *Permitir matrices dinámicas* en el apartado *Extensiones del lenguaje*.

Una vez cambiada la configuración, ya será posible la creación de los tipos de datos derivados.

Tipo de datos derivados creados	Tipo	Descripción
COMtablaENVIO	ARRAY[0...3] OF INT	Tipo de dato usado para la tabla de gestión de envíos.
COMtablaRECIBE	ARRAY[0...3] OF INT	Tipo de dato usado para la tabla de gestión de recepción.

4.3.6 Librería creada para la comunicación en Unity Pro XL

Con el fin de facilitar la posterior programación, se han desarrollado dos bloques para gestionar el algoritmo de redundancia. Los bloques creados son *Envio_word_con_CRC* y *Lectura_mensaje_recibido_Con_CRC*.

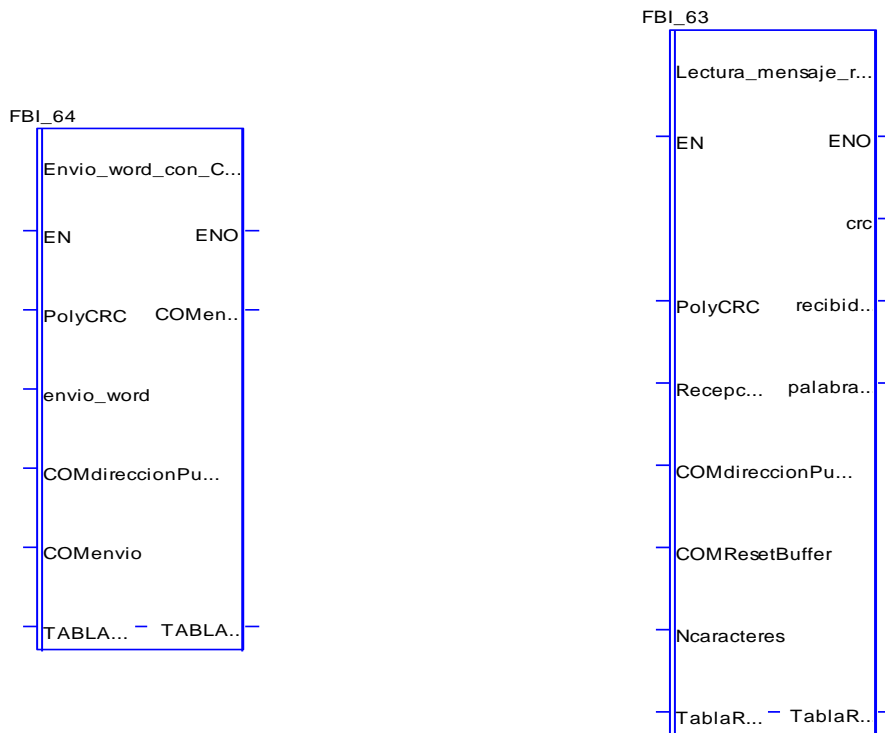


Ilustración 15: Bloques usados como librería básica.

A continuación, se verán con detalle las entradas y salidas de cada bloque.

4.3.6.1.1 Envio_mensaje_con_CRC

Calcula la cadena de caracteres a enviar partiendo del mensaje en formato WORD y el polinomio generador del CRC, y procede al envío de la cadena a la dirección indicada en los parámetros cuando se produzca un flanco de subida en la entrada *COMenvio*.

En la siguiente tabla se encuentran los parámetros de entrada y salida del bloque.

Tabla 13: Parámetros de entrada y salida de la función Envio_mensaje_con_CRC.

Parámetros	Entrada/salida	Tipo	Descripción
PolyCRC	Entrada	WORD	Polinomio CRC indicado en binario.
Mensaje	Entrada	STRING	Mensaje binario a enviar de 6 bits (en caracteres).
COMdireccionPuertoS	Entrada	ADDM_TYPE	Dirección del puerto serie.
COMenvio	Entrada	EBOOL	Variable que produce el envío con un flanco de subida.
TABLArecibe	Entrada/salida	COMtablaRECIBE	Tabla de gestión de intercambios
COMenviado	Salida	STRING	Muestra la cadena enviada por el puerto serie (incluyendo CRC).

4.3.6.2 Lectura_mensaje_recibido_Con_CRC

Recibe el mensaje recibido por el puerto serie en modo de caracteres y comprueba que la recepción del

mensaje ha sido correcta. Para ello, lleva a cabo el cálculo del algoritmo CRC.

Los parámetros de entrada y salida se encuentran en la Tabla 14. Con el fin de conocer si se ha recibido un nuevo mensaje se chequeará el valor de *TABLAenvio[3]*, que indicará el número de caracteres que contiene la cadena esperando a ser leída en el buffer.

Tabla 14: Parámetros de entrada y salida de Lectura_mensaje_recibido_Con_CRC.

Parámetros	Entrada/salida	Tipo	Descripción
PolyCRC	Entrada	WORD	Polinomio CRC indicado en binario.
Recepcion	Entrada	EBOOL	Si se encuentra a 1, entonces estará recibiendo.
COMdireccionPuertoS	Entrada	ADDM_TYPE	Dirección del puerto serie.
COMRessetBuffer	Entrada	INT	Si su valor es 1 resetea el buffer.
Ncaracteres	Entrada	INT	Número de caracteres a leer como máximo del puerto serie.
TABLAenvio	Entrada/salida	COMtablaENVIO	Tabla de gestión de intercambios.
CRC	Salida	WORD	CRC del mensaje recibido.
Recibido_correcto	Salida	EBOOL	Si está a 1 indica recepción de mensaje con CRC coherente.
Mensaje_recibido	Salida	WORD	Mensaje recibido con CRC. Se encuentra desplazado hacia la izquierda.

4.4. Programación del almacén ASRS

Para llevar a cabo la programación del almacén se ha empleado el lenguaje de programación ACL, siguiendo los manuales de referencia [5] y [10, 2]. Es necesario mencionar cómo se ha organizado la distribución de los estantes disponibles en el almacén en función del contenido de la pieza.

En el puesto robotizado de almacenaje de pallet se distingue entre pallets con contenido defectuoso, pallets con contenido sin tratar, y pallets con contenido procesado. Cada una de estas categorías tienen asignadas unos estantes, que se muestran en la Tabla 15. Los compartimentos sombreados en gris corresponden al almacenamiento de pallets con contenido sin procesar, los verdes a pallets con contenido procesado, y los rojos a pallets con contenido defectuoso.

4.4.1 Librería de bajo nivel

Con el fin de gestionar la recepción y el envío de caracteres por el puerto serie, se han usado dos funciones básicas: *GETCOM* y *SENCOM*. Toda la información sobre las funciones que se presenta a continuación ha sido extraída de los manuales de referencia citados anteriormente.

Tabla 15: Almacenamiento según su categoría.

Compartimentos del 31 al 36	Compartimentos del 67 al 72
Compartimentos del 25 al 30	Compartimentos del 61 al 66
Compartimentos del 19 al 24	Compartimentos del 55 al 60
Compartimentos del 13 al 18	Compartimentos del 49 al 54
Compartimentos del 7 al 12	Compartimentos del 43 al 48
Compartimentos del 1 al 6	Compartimentos del 37 al 42

4.4.1.1 GETCOM *n var*

Recibe un byte del puerto RS232 *n* especificado, y almacena su código ASCII en la variable *var* indicada. En este caso, el puerto serie usado ha sido el 0.

4.4.1.2 SENCOM *n arg1*

Envía un byte almacenado en la variable *arg1* en código ASCII por el puerto RS232 *n* especificado.

4.4.2 Librería creada para la comunicación del almacén.

4.4.2.1 Programa RECEI

Este programa se encargará de la recepción de una petición del PLC al almacén por el puerto RS232 0. Indicará la finalización de la ejecución poniendo la variable global FIN a 1, mientras que los errores de recepción serán indicados poniendo la variable ERREC a 1. La cadena de caracteres leída tendrá como máximo 12 caracteres (contando con los caracteres especiales que indican el final de la cadena a leer), siendo almacenada en la cadena MENSA[20] definida como global. En el caso de tener más caracteres, se indicará la existencia del error con la variable ERREC.

4.4.2.2 Programa CRC

Este programa se encarga del cálculo del CRC del mensaje recibido de forma que, si se realiza el CRC al mensaje ampliado y se obtiene resto 0, la recepción del mensaje ha sido correcta. Los parámetros de entrada y salida se encuentran descritos en la Tabla 16.

Tabla 16: Parámetros de entrada y salida del programa CRC.

Nombre del parámetro	Tipo	Descripción
MENAM[20]	Entrada	Mensaje ampliado: está formado por mensaje (6 caracteres) + CRC (4 caracteres) + 4 ceros (4 caracteres).
POLY[4]	Entrada	Es el polinomio usado para realizar el CRC.
FIN	Salida	Indica la terminación del programa.
CORRE	Salida	Si su valor es 1, indica que el mensaje recibido es correcto.
PETIC[4]	Salida	En el caso de que sea correcto, la petición sin el CRC se guarda en este vector.

4.4.2.3 Programa CRCEN

Dado el mensaje a enviar y el polinomio CRC, este programa calculará el CRC de un mensaje que se quiera enviar. Los parámetros de entrada y salida se encuentran descritos en la Tabla 17.

Tabla 17: Parámetros de entrada y salida del programa CRCEN.

Nombre del parámetro	Tipo	Descripción
MENSA[20]	Entrada	Mensaje ampliado: está formado por mensaje (6 caracteres) + CRC (4 caracteres) + 4 ceros (4 caracteres).
POLY[4]	Entrada	Es el polinomio usado para realizar el CRC.
FIN	Salida	Indica la terminación del programa.

4.4.2.4 Programa ENVAL

Se encarga del envío del vector de ocupación del sistema de almacenamiento de pallets. Con el fin de mantener la ocupación del almacén actualizada, cada vez que se lleve a cabo una tarea de extracción o almacenamiento de pallet en el almacén se deberá realizar el envío completo del vector de ocupación. La descripción de los parámetros de entrada y salida se encuentran en la Tabla 18.

Tabla 18: Parámetros de entrada y salida del programa ENVAL.

Nombre del parámetro	Tipo	Descripción
LLENO[72]	Entrada	Vector que contiene el estado de ocupación del almacén.
POLY[4]	Entrada	Es el polinomio usado para realizar el CRC.
FINAL	Salida	Indica la terminación del programa.

4.4.2.5 Programa PROB

Se encarga de responder todas las peticiones del almacén. Engloba todos los programas mencionados anteriormente.

Se usa un vector llamado SITUA que indica el estado actual del sistema robotizado de almacenaje. Los posibles estados son los mensajes que pueden ser enviados por éste al PLC principal (se encuentran en la Tabla 8). En el caso de solicitar el estado actual se enviará este vector, que contiene la información necesaria solicitada.

Este estado será actualizado por cada programa que lleve a cabo una petición (EXTRS, EXTRD, EXTRA, METE, METES y METED, que se explican a continuación).

4.4.2.6 Programas EXTRA, EXTRS y EXTRD

Realizan las peticiones de extracción de pallet con contenido procesado, contenido sin tratar, y contenido defectuoso respectivamente.

En primer lugar, realiza una búsqueda de un pallet que cumpla las características. En el caso de que no lo cumpla, actualizará el estado del almacén indicando error en la realización de la tarea. Una vez encontrado el pallet, se procede a la extracción del estante y al depositado en la bandeja ubicada en el retenedor del puesto de trabajo. Por último, se indica del finalizado del trabajo en el recurso compartido (la bandeja retenida en el sistema de transporte) con un cambio de estado.

4.4.2.7 Programas METE, METES, METED

Son análogos a los descritos anteriormente. Realizan las peticiones de almacenamiento de pallets con contenido procesado, sin tratar y defectuoso respectivamente. Su forma de procesar es completamente idéntica a la descrita en el sub-apartado anterior.

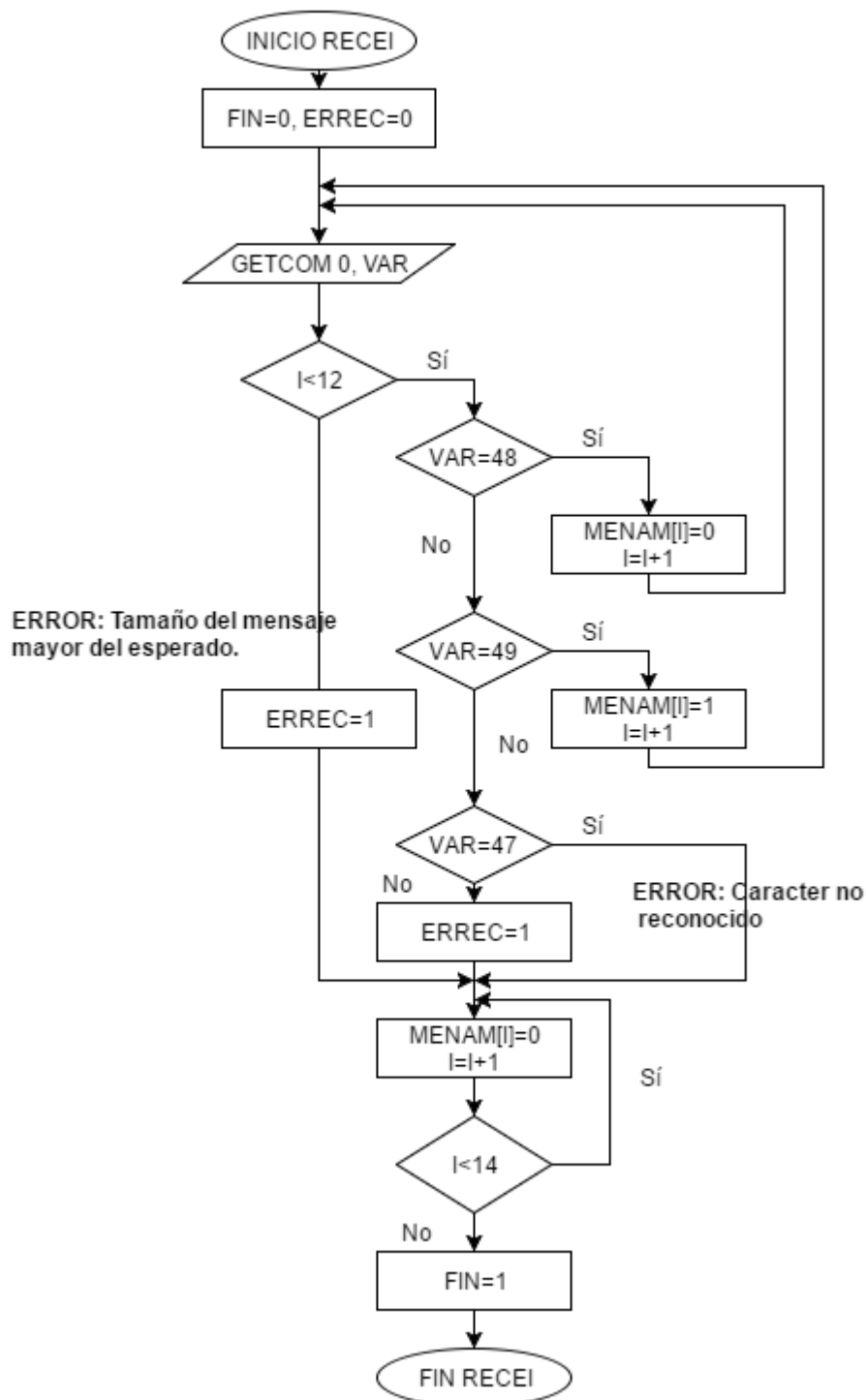


Ilustración 16: Diagrama de flujo del programa RECEI.

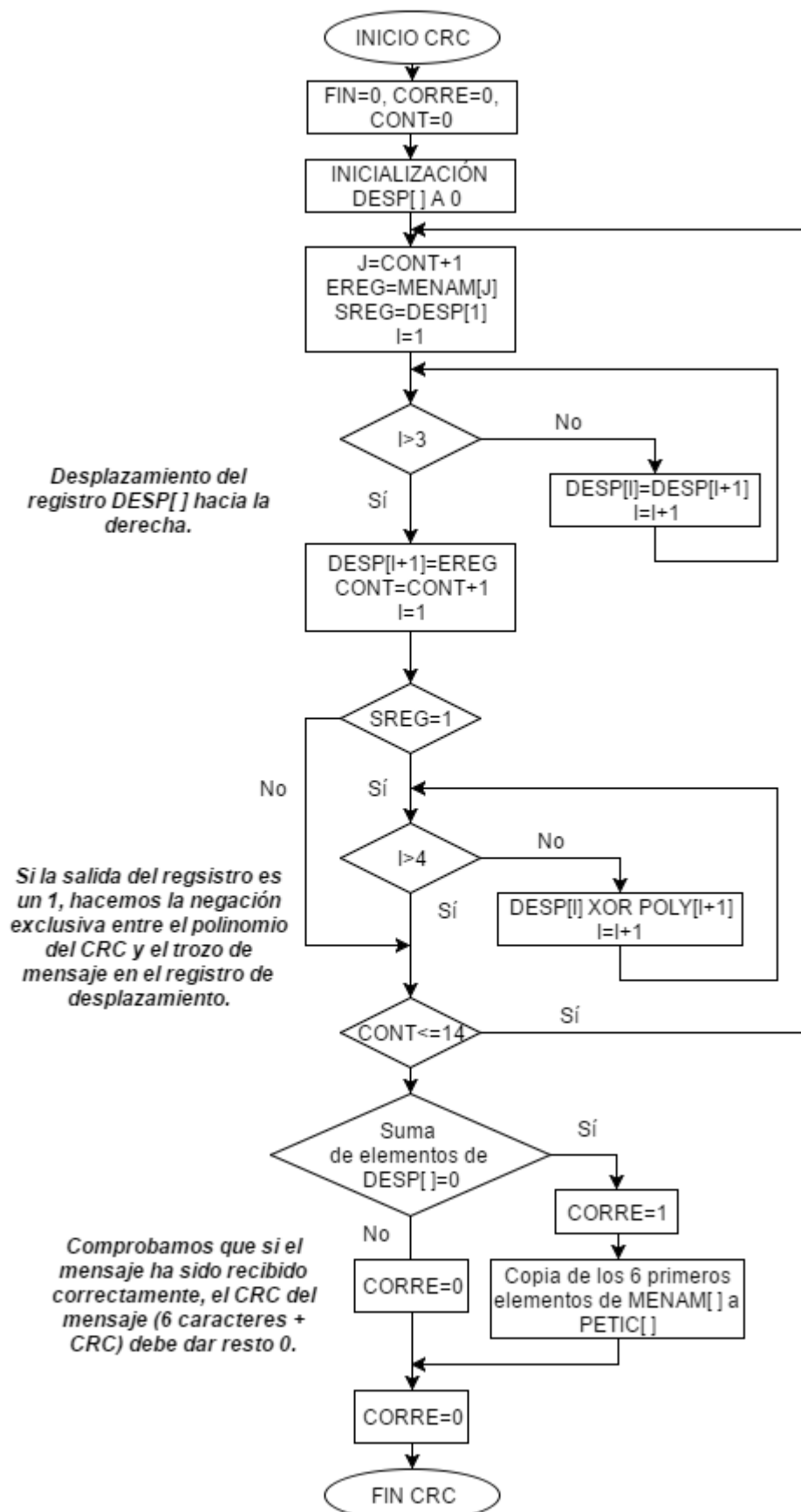


Ilustración 17: Diagrama de flujo del programa CRC.

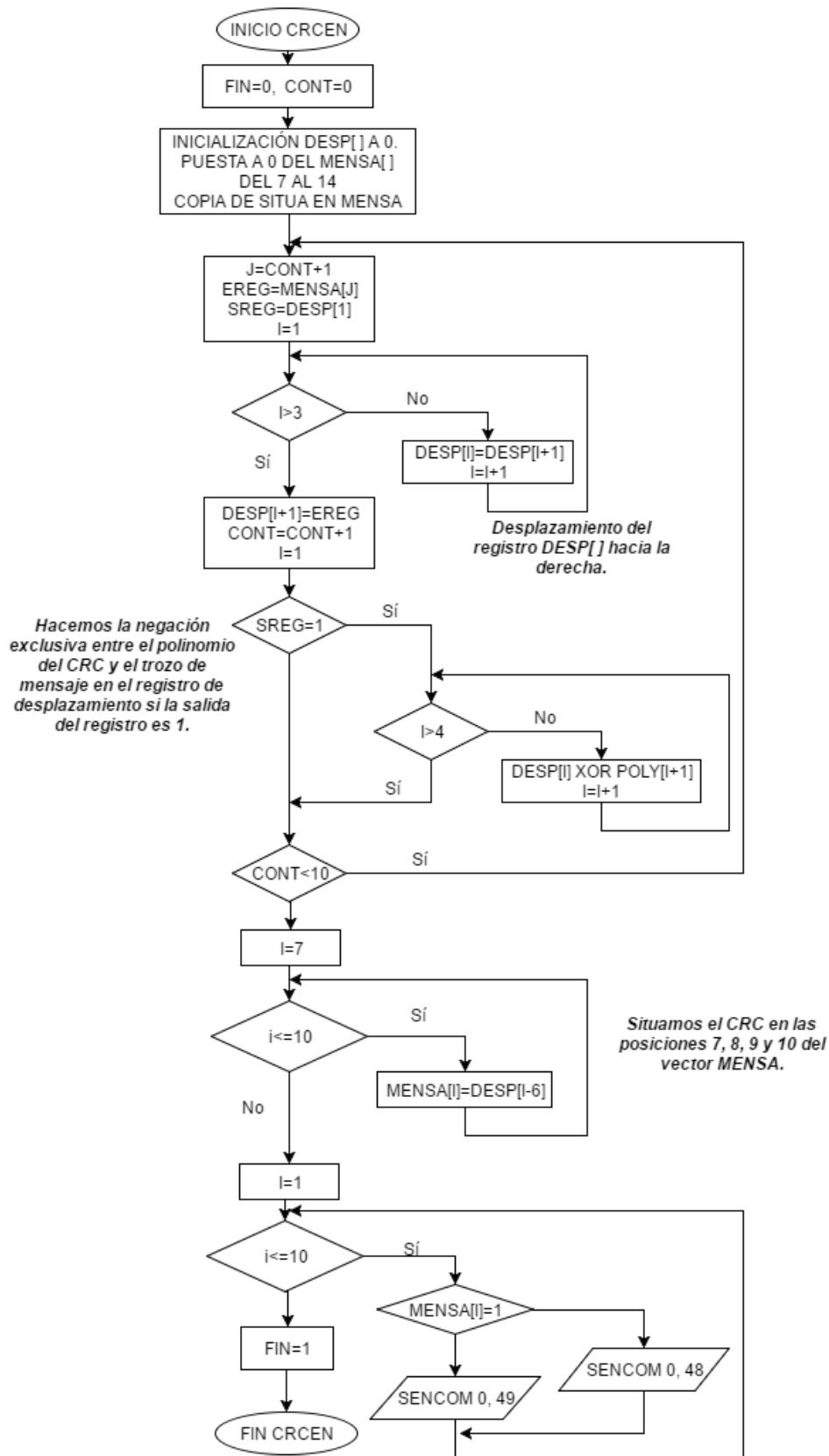


Ilustración 18: Diagrama de flujo del programa CRCEN.

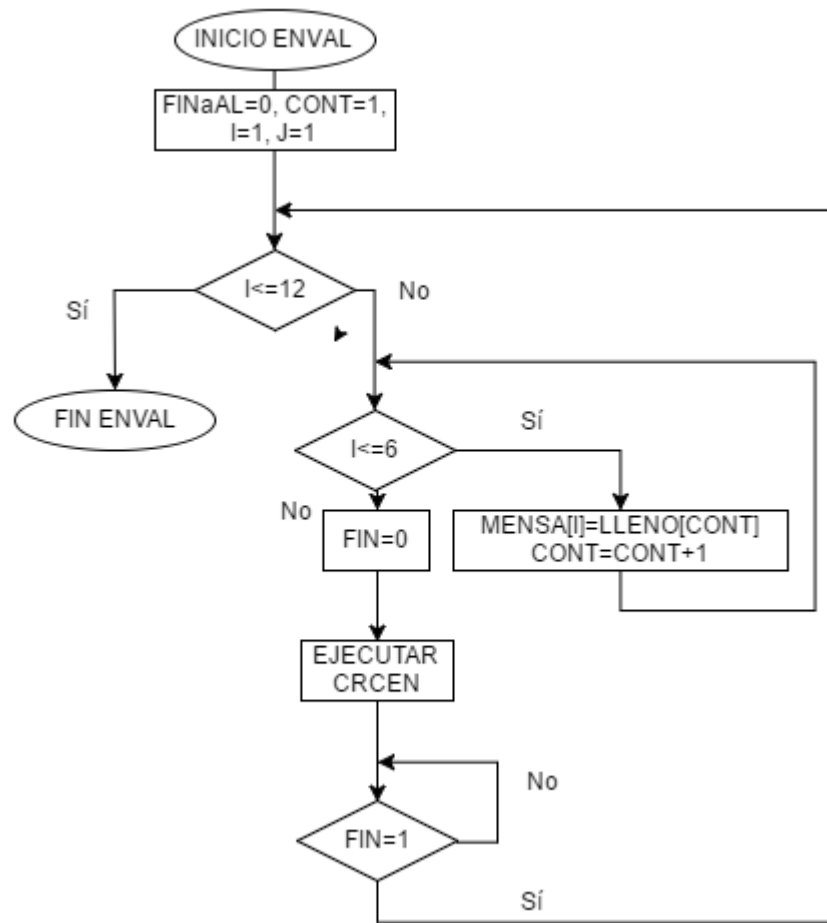


Ilustración 19: Diagrama de flujo del programa ENVAL.

5 COMUNICACIÓN ENTRE AUTÓMATAS

El autómata principal (aquél que se encarga de coordinar todos los puestos de trabajo que constituyen la instalación) deberá comunicarse con el resto de los autómatas de las distintas máquinas de la instalación.

En el caso de la célula de fabricación flexible, se disponen de diversos equipos equipados con autómatas:

- Alimentador de bandejas.
- Robot SCORBOT ER 4-u de Intelitek.
- Robot SCARA de Sony.
- Pórtico situado en el desvío 3 (en proceso de instalación).

Todos los autómatas de la instalación poseen el mismo módulo procesador (módulo encargado de las conexiones), por lo que se especificará cómo realizar la conexión del PLC principal con el autómata del alimentador de bandejas, siendo el procedimiento de conexión extrapolable al resto de los autómatas. Para la comunicación entre autómatas se ha usado el protocolo MODBUS sobre conexión Ethernet.

En esta comunicación el autómata principal tendrá acceso a las direcciones físicas de la memoria del autómata secundario, pudiendo leer o escribir datos sobre ella. En el caso de la máquina alimentadora de bandejas, se escribirán de forma remota las peticiones de almacenamiento o extracción.

5.1. Conexión del hardware

Los distintos equipos del laboratorio se encuentran interconectados en una red local Ethernet usando un conmutador Ethernet TCP/IP modelo TCSESU083FN0 de la gama de productos ConneXium del fabricante Schneider Electric. Éste se encuentra ubicado en el cuadro de mandos del sistema de transporte (junto al autómata principal).



Ilustración 20: Ubicación y detalle del conmutador [11].

Tabla 19: Características técnicas del conmutador disponible en la instalación.

Gama de producto	ConneXium
Protocolo de comunicación	Ethernet TCP/IP
Puertos Ethernet	10BASE-T/100BASE-TX - 8 puertos para cobre
Número de conmutadores en cascada	Ilimitado

En la Ilustración 22 se encuentra el aspecto de la CPU BMX P34 2020, en el cual se encuentra indicado el puerto Ethernet como el número 3.

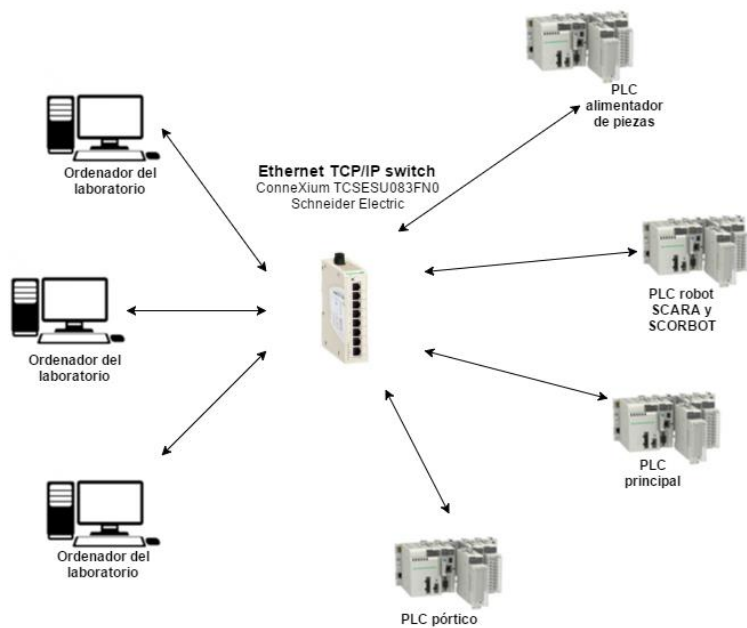


Ilustración 21: Esquema simplificado de la red local.

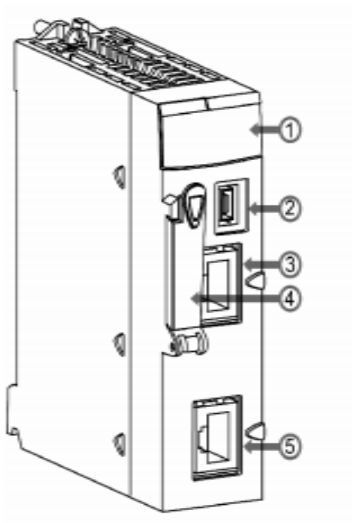


Ilustración 22: Ubicación del puerto Ethernet (3) en la CPU BMX P34 2020.

Tabla 20: Descripción de los puertos de la CPU BMX P34 2020.

1	Pantalla de indicadores LED
2	Puerto USB
3	Puerto Ethernet
4	Ranura de la tarjeta de memoria SD
5	Puerto serie MODBUS (conector RJ45)

5.2. Configuración de la red Ethernet en autómatas usando Unity Pro XL

Una vez configurados los módulos de cada autómata concreto en el apartado Configuración del *Explorador de proyectos*, se deberá configurar la conexión Ethernet. Para ello, en el *Explorador de proyectos* se hace click en el botón derecho en el directorio *Redes* (dentro de la carpeta *Comunicación*) y se selecciona *nueva red*.

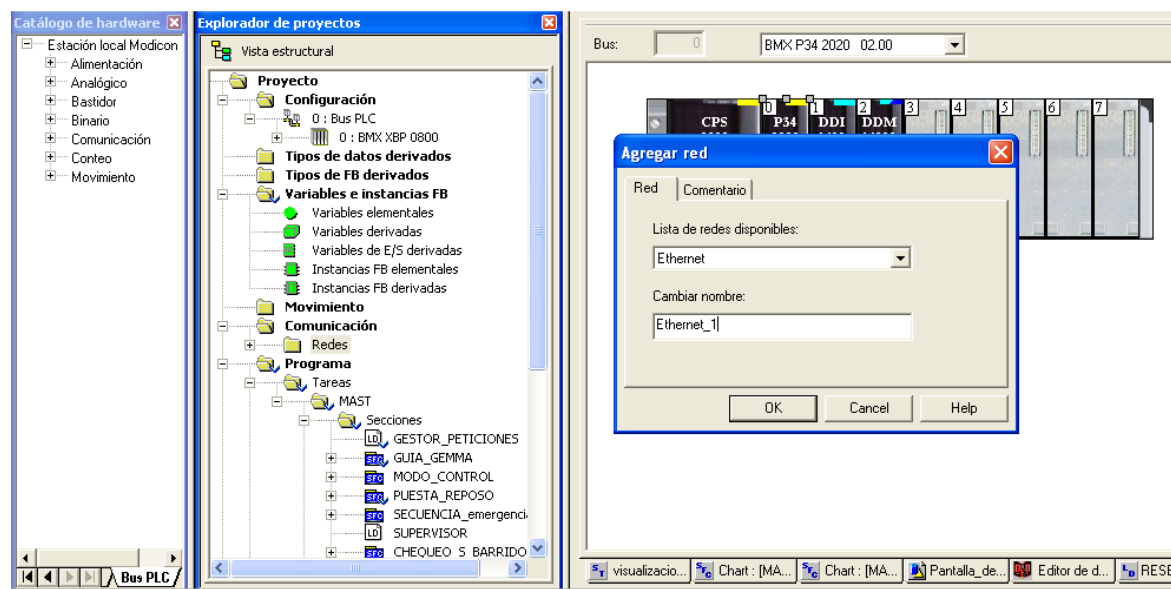


Ilustración 23: Ventana de configuración de nueva red Ethernet.

Tras elegir un nombre para la red, se pulsa en OK, añadiendo una nueva red Ethernet en la carpeta de Redes. Pulsando en ella se accederá a la ventana que permita la configuración de la dirección IP del equipo:

- Dirección IP: Dependerá del equipo a conectar. En el caso de la célula de fabricación flexible, las direcciones son:
 - Autómata del alimentador de bandejas 192.168.0.15
 - Autómata principal 192.168.0.12
 - Autómata conectado al robot SCARA y SCORBOT-ER: 192.168.0.16
 - Autómata conectado al pórtico: 192.168.0.14
- Máscara de subred: 255.255.255.0
- Dirección de pasarela: 0.0.0.0

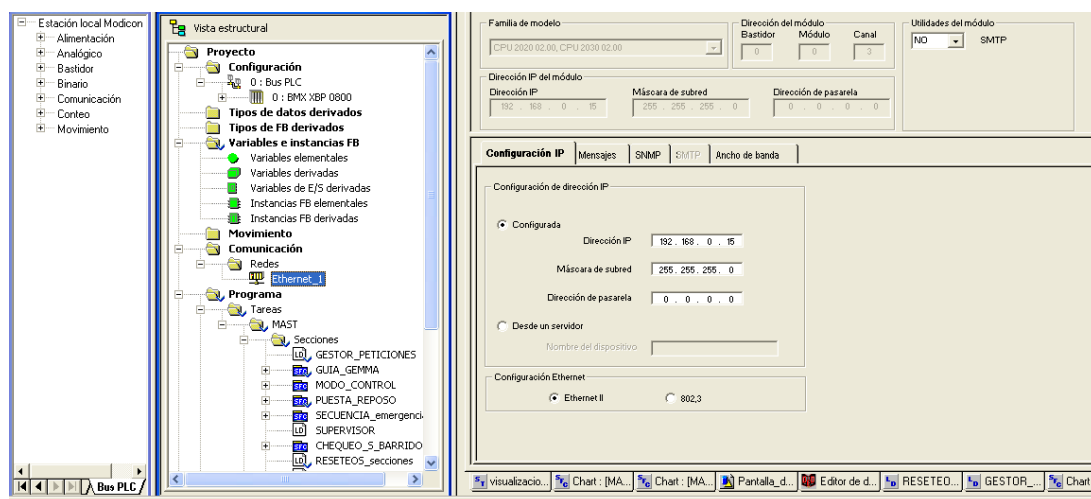


Ilustración 24: Ventana de configuración de la dirección IP de la red Ethernet.

Tras la configuración, es importante que se cierre la ventana con el fin de validar la configuración realizada. Una vez validado, será necesario abrir la ventana de configuración a partir del Explorador de proyectos, y en ella, pulsar sobre la entrada Ethernet del módulo procesador. En la ventana de configuración emergida, se deberá comprobar en el margen inferior izquierdo que la función sea ETH TCP IP y que la conexión de red sea la agregada anteriormente.



Ilustración 25: Detalles de la configuración de la red Ethernet.

5.3. Librería básica usada

Con el fin de llevar a cabo estas lecturas o escrituras de variables en el PLC secundario, se han usado los bloques *ADDM*, *READ_VAR* y *WRITE_VAR*.

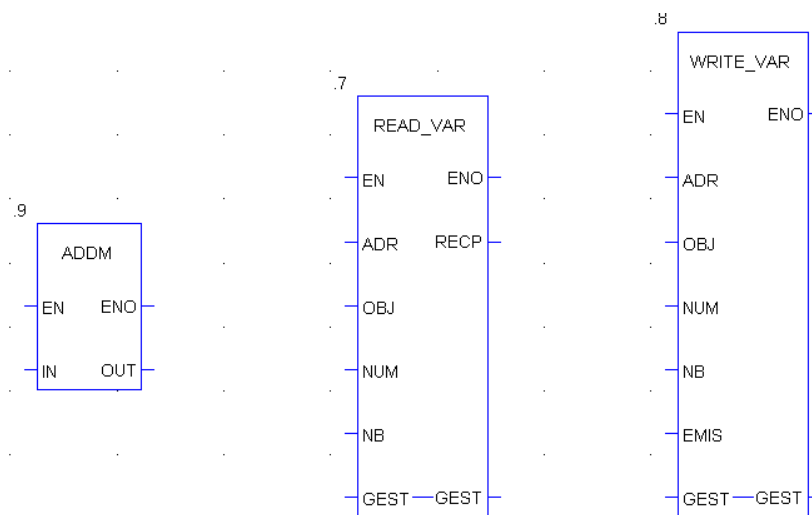


Ilustración 26: Bloques usados en la comunicación por MODBUS.

5.3.1 ADDM

Este bloque ya fue explicado anteriormente en el apartado de la conexión del equipo robotizado de almacenaje de pallets con el autómata principal (apartado 4.3.4.1).

5.3.2. READ_VAR

Permite la lectura de bit y palabras internas entre autómatas. Los parámetros de entrada y salida se encuentran especificados en la siguiente tabla.

Tabla 21: Parámetros de entrada y salida de READ_VAR.

Parámetros	Entrada/Salida	Tipo	Descripción
ADR	Entrada	ADDMM	Dirección del PLC a leer generada por la función ADDMM.
OBJ	Entrada	STRING	En caso de lectura de tipo EBOOL, se usará '%M', en caso de palabras '%MW'.
NUM	Entrada	DINT	Número del primer elemento a leer.
NB	Entrada	INT	Número de objetos a leer.
GEST	Entrada/salida	ARRAY[0..3 OF INT]	Tabla de gestión de la comunicación.
RECP	Salida	ANY_ARRAY_INT	Variable donde se almacenan los valores leídos.

Es necesario hacer hincapié en el caso de la lectura de bits internos, ya que el bloque almacena la lectura en un tipo INT. Por ello es necesario el cambio de tipo INT a WORD usando el bloque INT_TO_WORD, y luego descomponer el WORD obtenido en bits usando el bloque WORD_TO_BIT.

5.3.3. WRITE_VAR

Permite la escritura de bits y palabras internas entre autómatas. La descripción de los parámetros de entrada y salida se encuentran en la Tabla 22.

Tabla 22: Parámetros de entrada y salida de WRITE_VAR.

Parámetros	Entrada/Salida	Tipo	Descripción
ADR	Entrada	ADDMM	Dirección del PLC a leer generada por la función ADDMM.
OBJ	Entrada	STRING	En caso de escritura de tipo EBOOL, se usará '%M', en caso de palabras '%MW'.
NUM	Entrada	DINT	Número del primer elemento a leer.
NB	Entrada	INT	Número de objetos a escribir.
EMIS	Entrada	ANY_ARRAY_INT	Valores a tomar por la palabra/bit a escribir.
GEST	Entrada/salida	ARRAY[0..3 OF INT]	Tabla de gestión de la comunicación.

6 COMUNICACIÓN DE AUTÓMATAS CON EL SCADA

En la siguiente sección se explorarán distintas opciones de conexionado y comunicación del autómatas con el sistema de supervisión SCADA, desarrollado con el programa Vijeo Citect de Schneider Electric. También se hará una comparación entre las distintas opciones.

Las distintas opciones de comunicación que se verán son mediante OPC (OLE for Process Control), comunicación general, o mediante conexión Unity SpeedLink, comunicación propietaria entre equipos de Schneider Electric.

La breve introducción teórica de la conexión OPC está basada en la guía teórica introductoria al OPC [12].

6.1. Conexión OPC

El OPC es un estándar de comunicación que permite la comunicación de dispositivos, controladores y aplicaciones sin necesidad de conocer los protocolos de comunicación nativos y la organización interna de los datos de cada uno de ellos. Esta conexión se realiza a través de una arquitectura cliente-servidor, formada por un cliente OPC y un servidor OPC.

El servidor OPC es una aplicación de software que hace de vínculo entre los protocolos nativos de los dispositivos o aplicaciones que hacen de origen de los datos, y el estándar de comunicación OPC. Las fuentes de datos que pueden ser comunicadas por un servidor OPC abarcan muchos tipos: autómatas, dispositivos, instrumentos de medición, bases de datos, etcétera.

Los clientes OPC son módulos de software usados por cualquier aplicación para comunicarse con cualquier servidor OPC. Conceptualmente, traducen las peticiones de comunicación procedentes de las aplicaciones o dispositivos en la petición OPC equivalente, y las comunican al servidor OPC correspondiente para que sea procesada. Éstos se encuentran implementados en aplicaciones como HMIs ³, SCADAs, generadores de informes, etcétera.

La comunicación OPC permite el uso de variables elementales sin necesidad de que se encuentren direccionadas de forma física en el autómatas.

6.1.1 Procedimiento de conexión del autómatas al SCADA

Se tratará el procedimiento de conexión para el caso del alimentador de bandejas, cuyo autómatas ha sido programado en Unity Pro XL con el SCADA desarrollado en el programa Vijeo Citect versión 7.0r3. Este procedimiento será extrapolable a la conexión de cualquier otro autómatas de los que se encuentran en la instalación tratada.

³Interfaz hombre-máquina (Human Machine Interface).

6.1.1.1 Exportación de variables en Unity Pro XL

Se exportan las variables necesarias pulsando la tecla derecha del ratón sobre las variables deseadas, y pulsando en exportar. El formato necesario será *‘.XVM’*. Para poder realizar la exportación, es necesario estar desconectado del autómata.

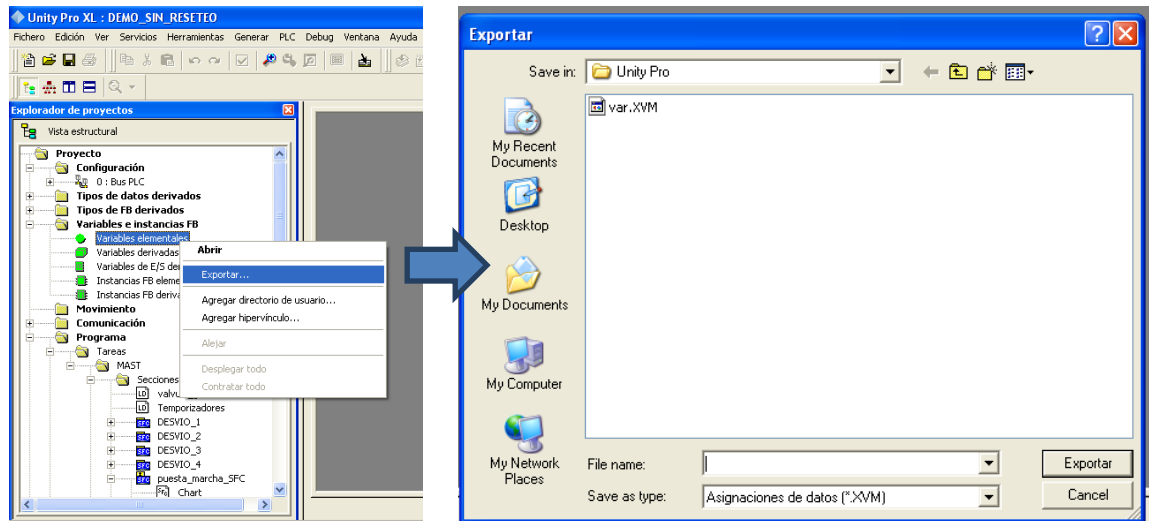


Ilustración 27: Exportación de variables en Unity Pro XL.

6.1.1.2 Configuración del servidor OPS

Se ejecuta la herramienta OFS Configuration Tool, disponible en el directorio de Schneider Electric (se puede acceder usando el menú inicio). En ella, se creará un nuevo *Alias*.

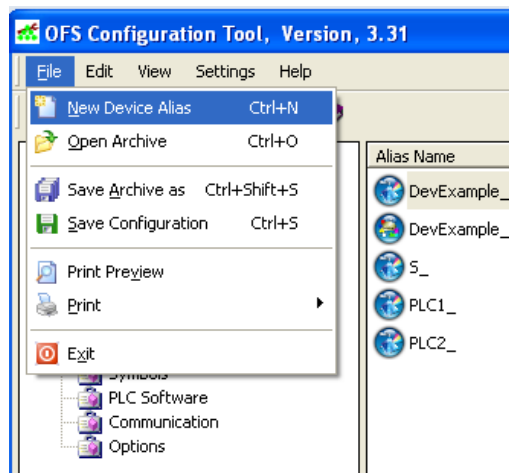


Ilustración 28 Creación del nuevo Alias.

Tras crear el alias, se pulsa sobre él para acceder a la ventana de configuración, y en ella se pulsa sobre la dirección del dispositivo (*Device Address*). En la ventana de configuración, se indica la dirección del dispositivo, y el programa usado en su programación (UNITY en este caso). Este paso se encuentra ilustrado en Ilustración 29.

Ahora se indica el directorio donde se encuentran las variables exportadas, que hacen de base de datos del servidor. Para ello, en la pantalla de las propiedades del Alias, se pulsa en *Symbol table file* (indicado con una flecha azul en Ilustración 30).

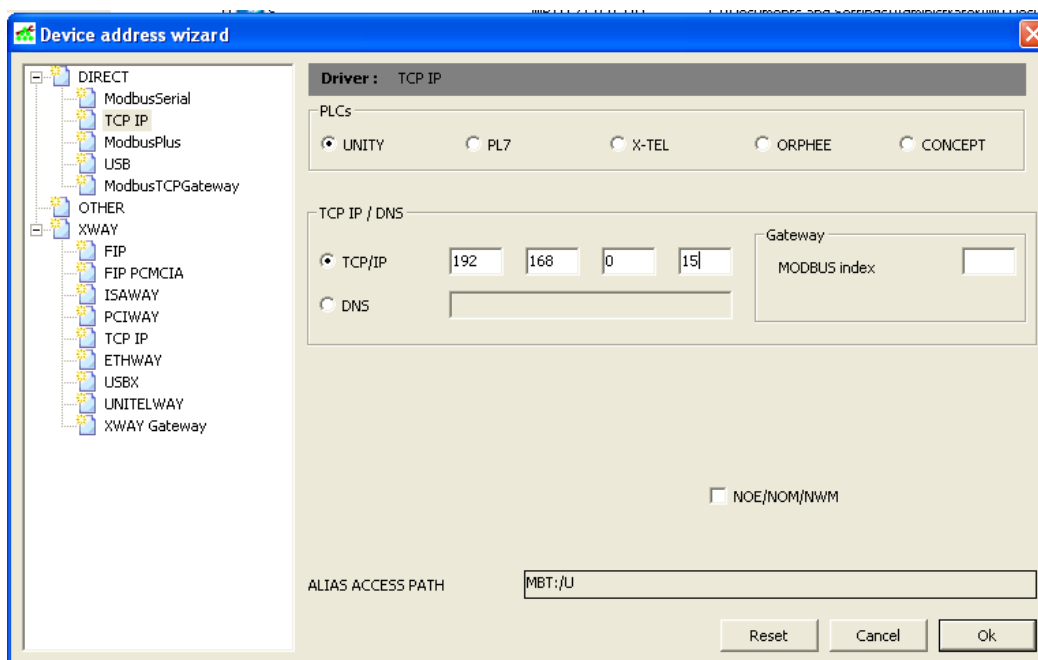


Ilustración 29: Ventana de configuración de la dirección.

Antes de validar los cambios, es necesario pulsar en la casilla de *Device* (en el apartado *Preload Settings*) para que la precarga de los datos se realice desde el dispositivo, y comprobar que la casilla *Simulated* no se encuentra marcada (en el apartado *Options*).

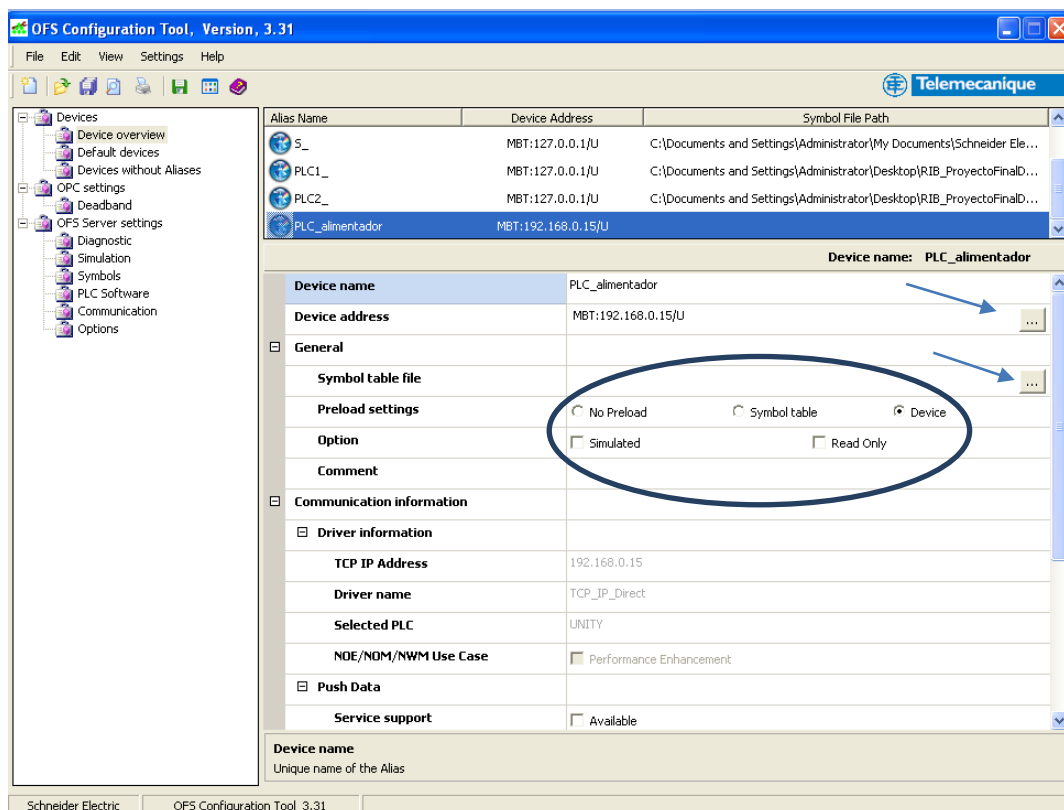


Ilustración 30 Configuración del Alias.

Por último, para validar los cambios se pulsa en *Save configuration* en el menú desplegable *File* (en la barra superior). Es necesario reiniciar el servidor en caso de que estuviera en ejecución.

Con el fin de comprobar la correcta configuración del servidor, se ejecutará un programa cliente OPC que se conectará a este servidor OPC. En este caso se ha usado OFSClient, un programa que se encuentra ubicado en el mismo directorio que la herramienta usada en el paso anterior.

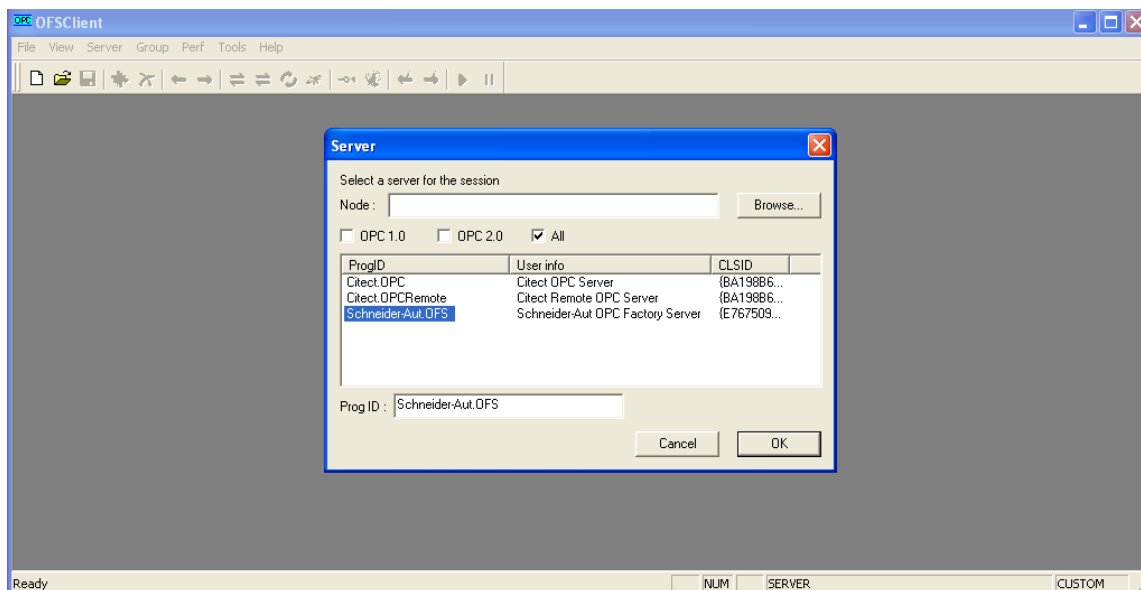


Ilustración 31: Elección del servidor OPC.

Una vez ejecutado, se cliquee en el servidor creado (Schneider-Aut.OFS) y se pulsa OK.

Ahora se pasa a crear un nuevo grupo de variables con el fin de comprobar el correcto funcionamiento del servidor. Para ello, se pulsa en *New*, ubicado en el menú desplegable *File*.

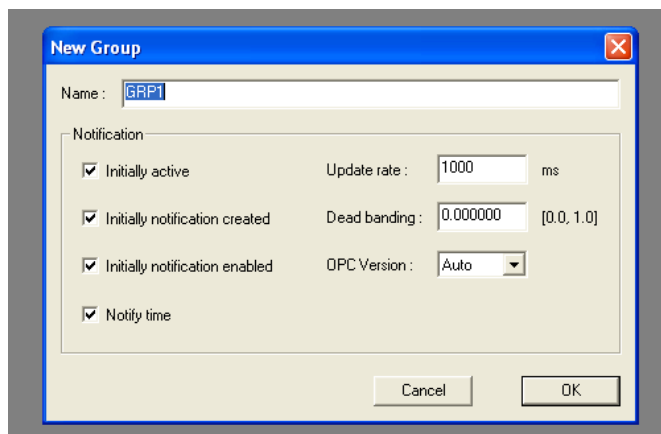


Ilustración 32: Creación del nuevo grupo de variables.

En la ventana se podrá elegir el nombre y la configuración del grupo a crear.

Una vez creado el grupo, se pasa a crear los distintos ítems que visualizarán el estado de las variables. Para ello, se pulsa en *New*, en el menú desplegable *Item* ubicado en la barra superior, dando acceso a la pantalla mostrada en la Ilustración 33.

En la ventana, se pulsa sobre el Alias creado anteriormente en el servidor, y se seleccionan aquellas variables que se deseen usar. Una vez hecho esto, se podrá ver si la conexión se ha establecido correctamente cambiando el valor de alguna de sus variables y viendo que el estado cambia en Unity Pro XL.

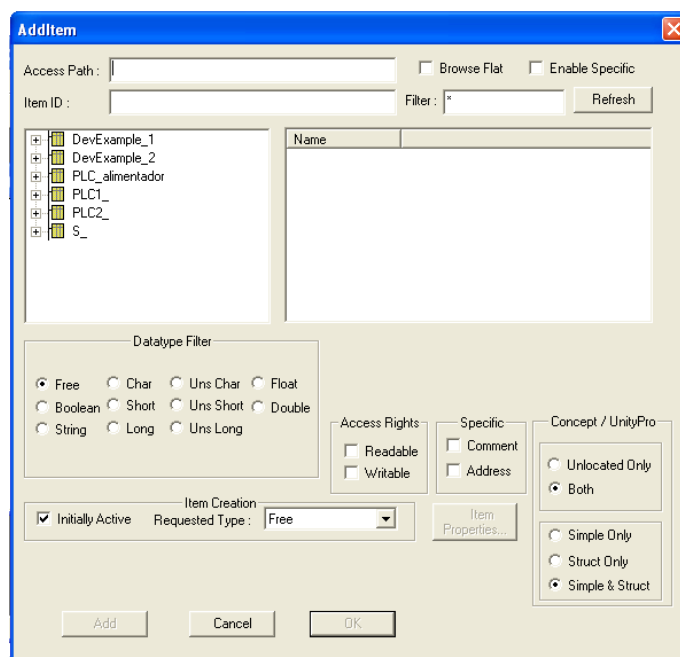


Ilustración 33: Ventana de creación de ítems.

En el caso de una conexión incorrecta, en la variable aparecerá un símbolo de advertencia en el margen derecho (como el mostrado en Ilustración 34).

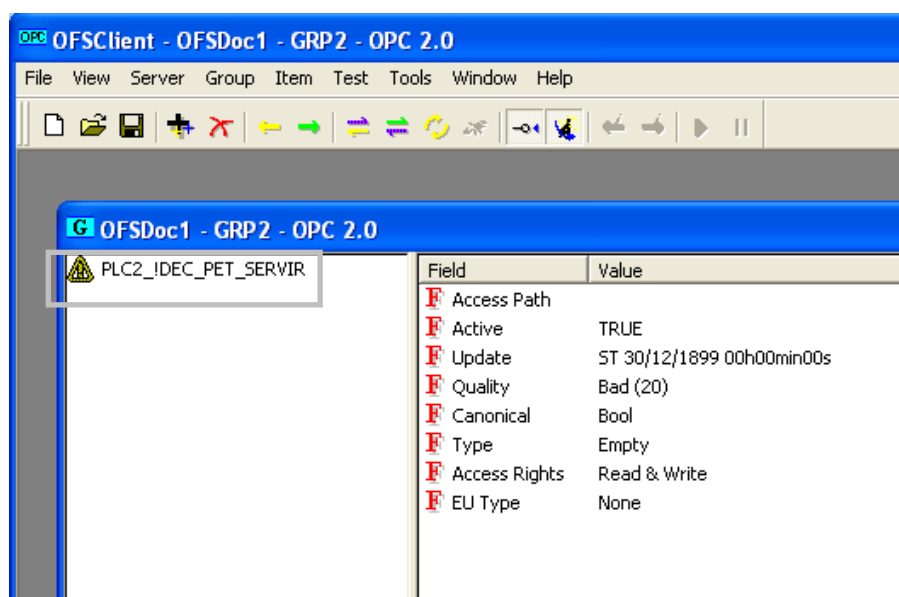


Ilustración 34: Ventana en el caso de una conexión incorrecta.

6.1.1.3 Configuración en Vijeo Citect SCADA

En primer lugar, se configura el dispositivo de E/S. Para ello, se pulsa en Configuración rápida de dispositivos de E/S, dentro del directorio *Comunicaciones*, que se encuentra en la parte derecha del explorador de proyectos.

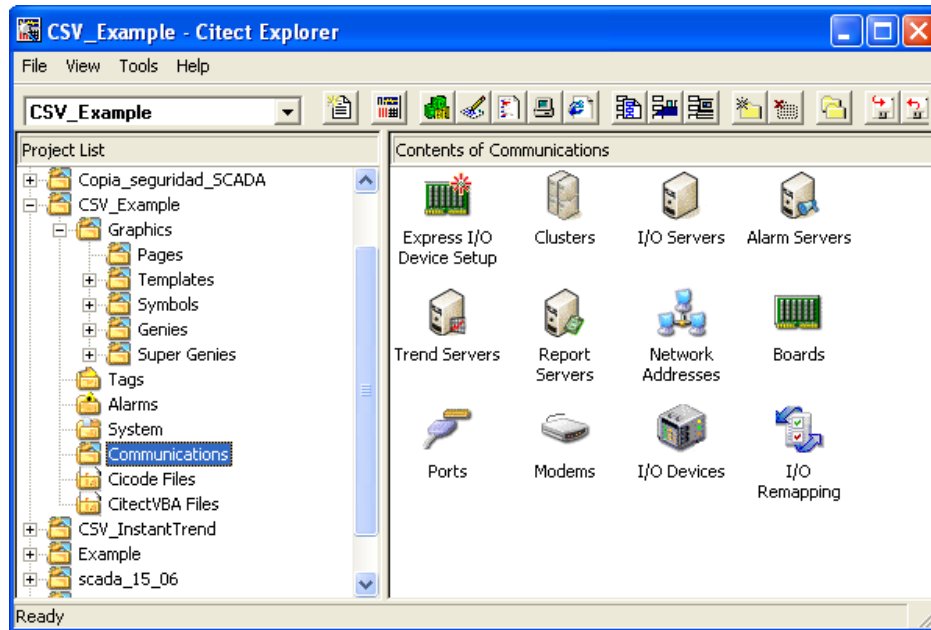


Ilustración 35: Explorador de proyectos de Vijeo Citect.

En la ventana de selección de servidor E/O, se creará un nuevo servidor.

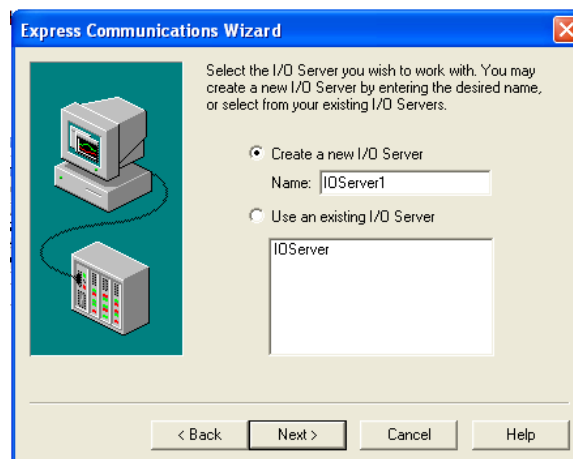


Ilustración 36: Creación de servidor E/S.

En la ventana de selección del tipo de selección de E/S, se selecciona Dispositivo de E/S externo, y se pulsa en siguiente (Ilustración 37).

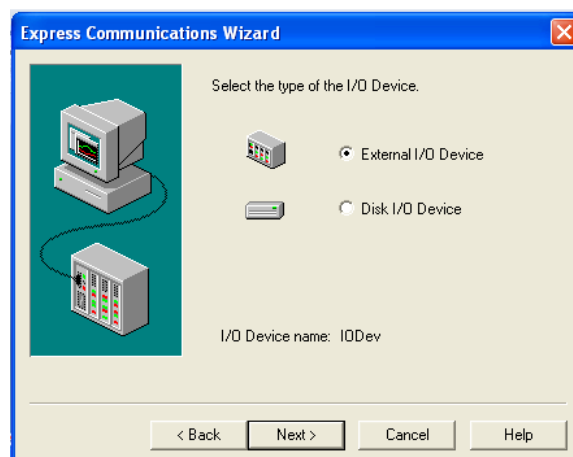


Ilustración 37: Selección de tipo de dispositivo.

En la ventana de método de comunicación del dispositivo de E/S (Ilustración 38), se selecciona OPC Factory Server, dentro del fabricante Schneider-Electric.

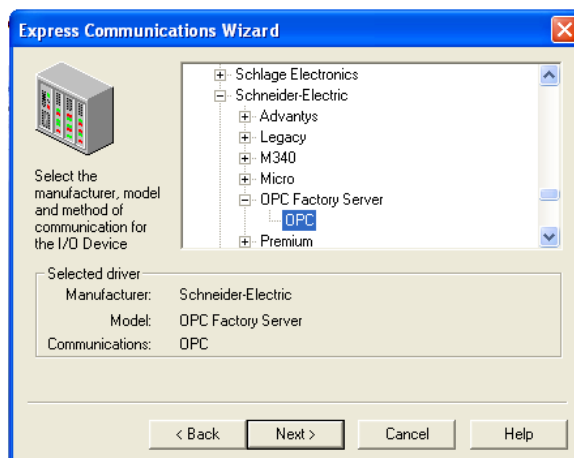


Ilustración 38: Selección del método de conexión.

A continuación, se introduce como dirección del dispositivo Schneider-Aut.OFS, y se pulsa siguiente (Ilustración 39).

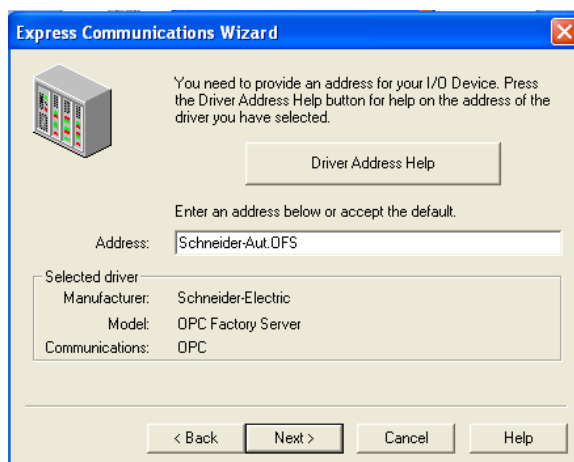


Ilustración 39: Selección de la dirección del servidor OPC.

En este caso no se ha usado una base de datos externa, por lo que no se debe marcar la casilla que aparece en el siguiente cuadro de diálogo:

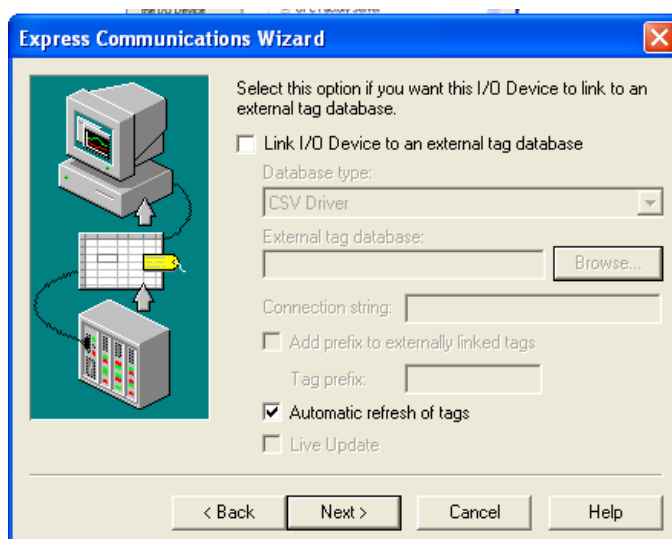


Ilustración 40: Opción de vincular a una base externa de datos de tags.

Por último, se pulsa en el resto de ventanas en siguiente hasta finalizar su creación.

Ahora, se procede a la creación del dispositivo de E/S. Para ello, en la ventana de comunicaciones del *Explorador de proyectos*, se pulsará en Dispositivos de E/S (Ilustración 41).

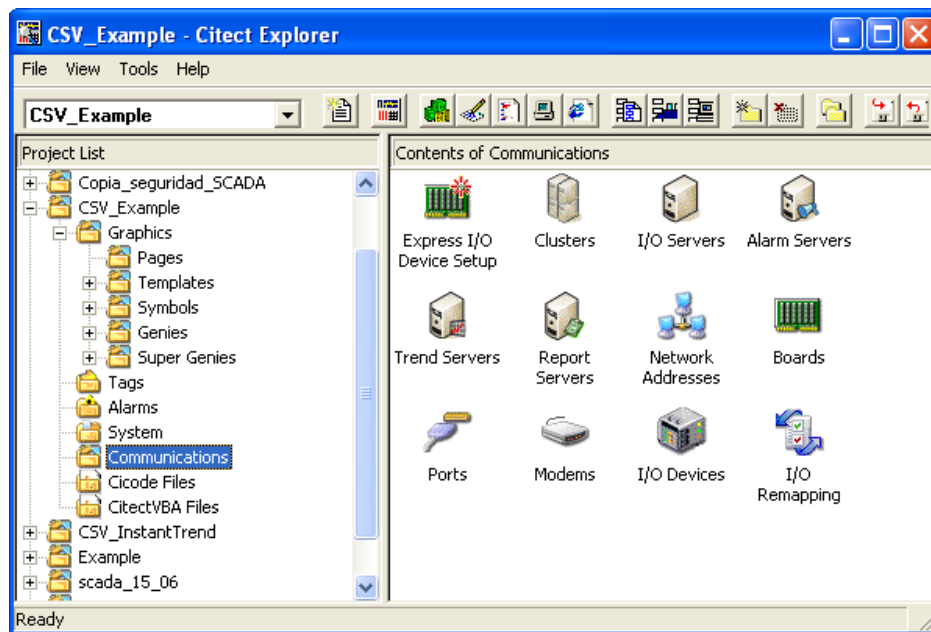


Ilustración 41: Apartado de comunicaciones del proyecto.

Una vez allí, se añade el nuevo dispositivo a conectar, indicando como dirección del dispositivo Schneider-Aut.OFS, y como protocolo OPC.

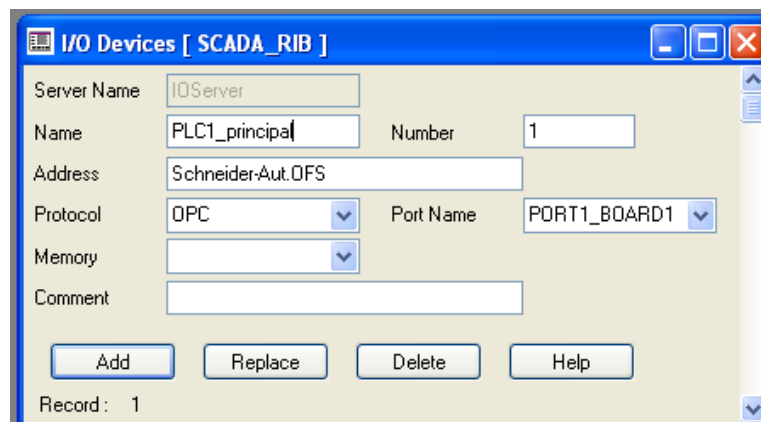


Ilustración 42: Creación de dispositivo de E/S.

Por último, es necesario realizar la exportación de tags de variables desde Unity pro XL, con el fin de escribir o leer variables en un dispositivo externo. Para ello, en el explorador de proyectos, se pulsará en *Importar tags*, en el menú desplegable *Herramientas* (Ilustración 43).

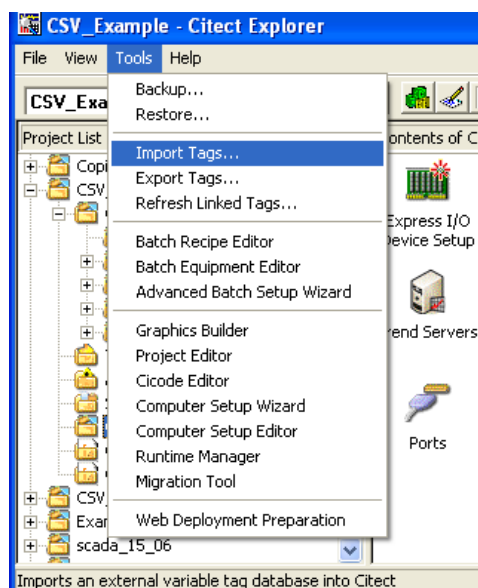


Ilustración 43: Importar tags desde el explorador de proyectos.

En la ventana de importación, se selecciona el dispositivo de E/S configurado anteriormente. En el tipo de base de datos se debe marcar OPC. También es necesario poner el directorio en el que se encuentran las variables exportadas desde Unity Pro XL (paso realizado anteriormente). Como opciones extras se encuentran el añadir un prefijo en la exportación de los tags, o el eliminado de los tags de este dispositivo de E/S antes de importar de nuevo.

Por último, se pulsa en *Importar* para proceder con la operación.

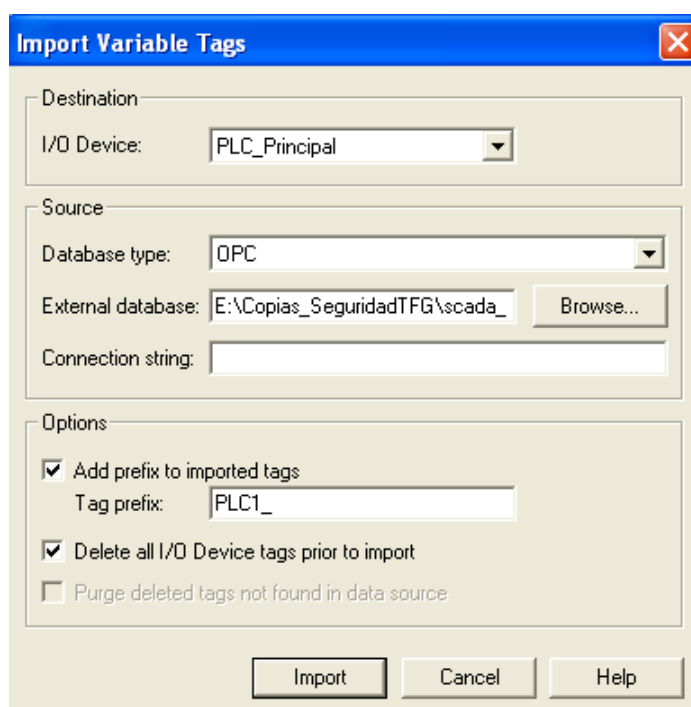


Ilustración 44: Importación de tags usando OPC.

6.2. Unity Speedlink Static

Unity Speedlink es una herramienta de comunicación incluida en Vijeo Citect usada para vincular un proyecto de Unity Pro XL con un proyecto de Vijeo Citect. Es un método de comunicación propiedad de Electric Schneider específico para equipos de este fabricante.

Existen dos tipos de comunicación:

- **Unity Speedlink Dynamic:** Establece un vínculo estático entre Vijeo Citect y un archivo de proyecto ‘.stu’. Todos los cambios que se realicen sobre las variables en Unity Pro XL no serán efectivos hasta que los tags sean actualizados manualmente.
- **Unity Speedlink Static:** Establece un vínculo dinámico entre Vijeo Citect y un archivo de proyecto ‘.XSY’. Todos los cambios que realizados sobre las variables en Unity Pro XL serán realizados en Vijeo automáticamente.

Unity Speedlink permite el uso de variables que tengan asignados una dirección de memoria (no siendo válidas aquellas que se encuentren direccionadas en los módulos de entrada o salida del automático). En Unity Pro XL, las palabras usan una dirección de tipo ‘%MW’, mientras que los bits usan una dirección de tipo ‘%M’.

A continuación, se explica el procedimiento de conexión.

6.2.1 Procedimiento de conexión del automático usando Unity Speedlink Static

6.2.1.1 Exportación de variables en Unity Pro XL

Se exportan las variables necesarias pulsando la tecla derecha del ratón sobre las variables deseadas, y pulsando en exportar. El formato necesario será ‘.XSY’. Para poder realizar la exportación, es necesario estar desconectado del automático.

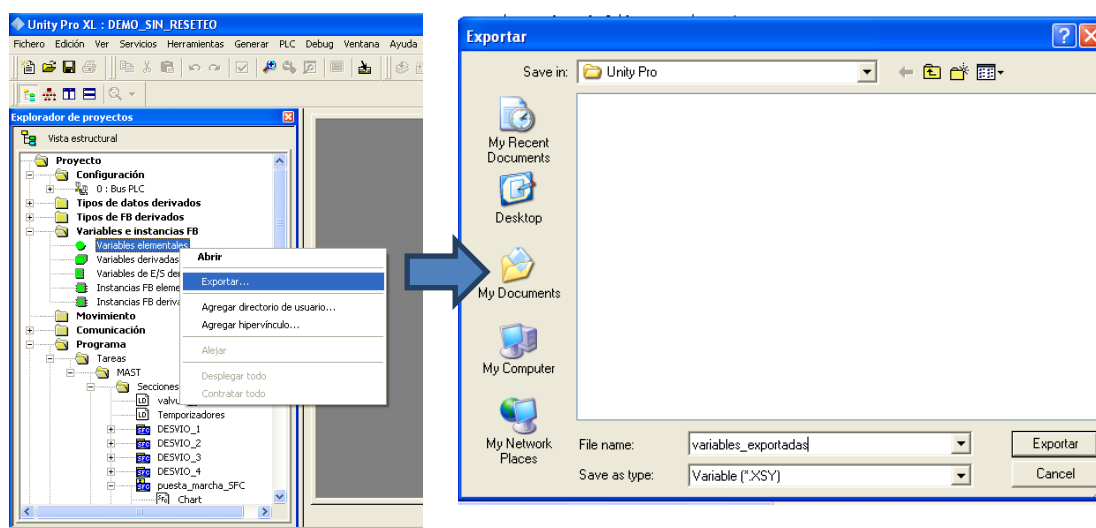


Ilustración 45: Exportación de variables en Unity Pro XL.

6.2.1.2 Configuración en Vijeo Citect SCADA

En primer lugar, se configura el dispositivo de E/S. Para ello, se pulsa en Configuración rápida de dispositivos de E/S, dentro del directorio Comunicaciones, que se encuentra en la parte derecha del explorador de proyectos.

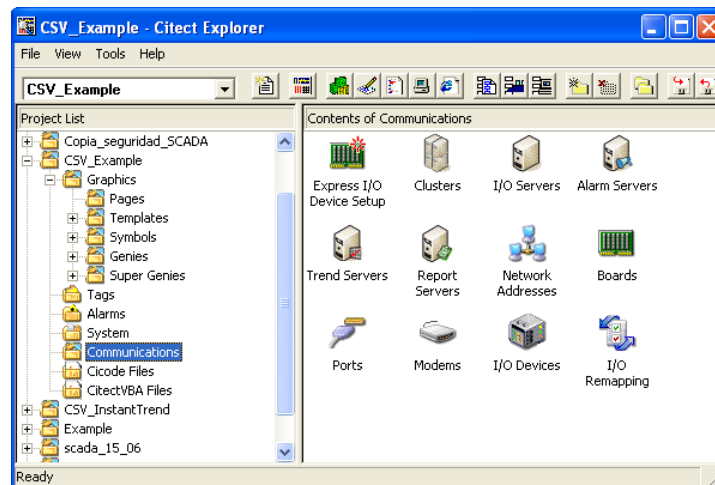


Ilustración 46: Apartado comunicaciones en el explorador de proyectos.

En la ventana de selección de servidor E/O, se creará un nuevo servidor.



Ilustración 47: Creación de nuevo servidor.

En la ventana de selección del tipo de selección de E/S, se selecciona Dispositivo de E/S externo, y se pulsa en siguiente.



Ilustración 48: Selección de tipo del dispositivo de E/S.

En la ventana de método de comunicación del dispositivo de E/S se busca Modbus/TCP (Ethernet), dentro de la gama M340 del fabricante Schneider-Electric.

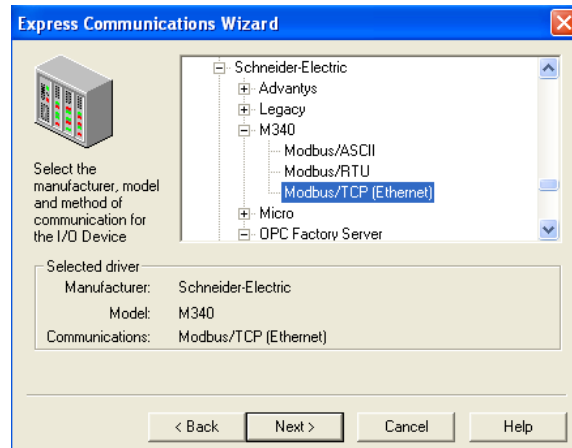


Ilustración 49: Selección de Modbus como método de comunicación.

A continuación, se introduce como dirección del dispositivo la dirección IP del autómata (en este caso el autómata principal), y como puerto 502.

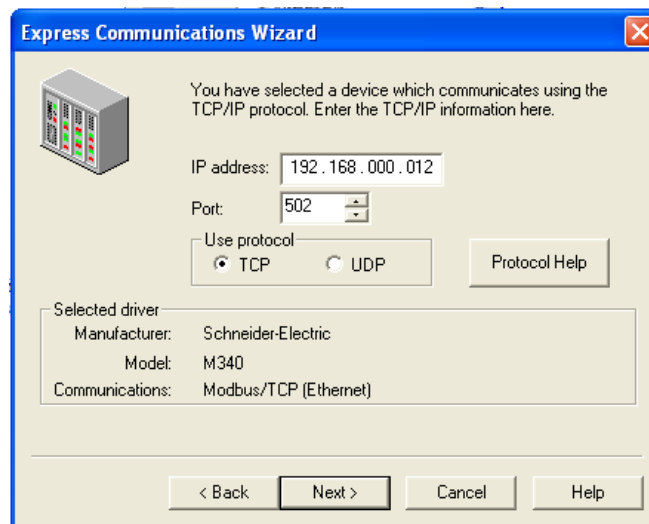


Ilustración 50: Selección de dirección IP y puerto.

En este caso no se ha usado una base de datos externa, por lo que no se debe marcar la casilla que aparece en el cuadro de diálogo de la Ilustración 51.

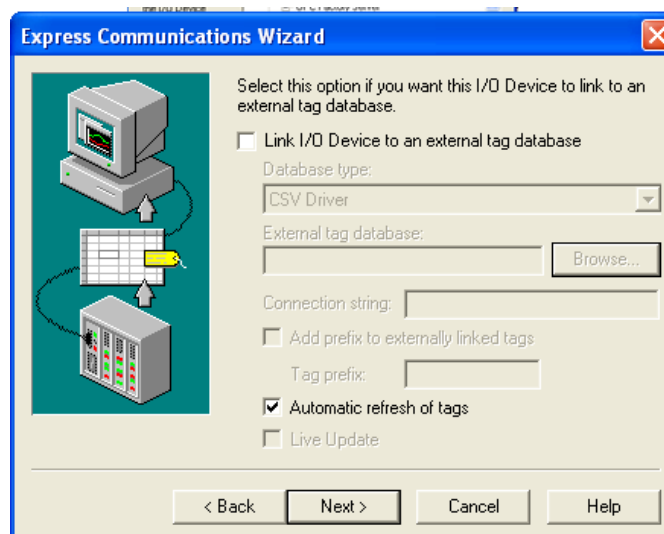


Ilustración 51: Posibilidad de vincular con una base externa de datos.

Por último, se pulsa en el resto de ventanas en siguiente hasta finalizar su creación.

Una vez realizada la configuración, se procede a crear el dispositivo de E/S. De nuevo, en la ventana de comunicaciones del *Explorador de proyectos*, se pulsará en Dispositivos de E/S.

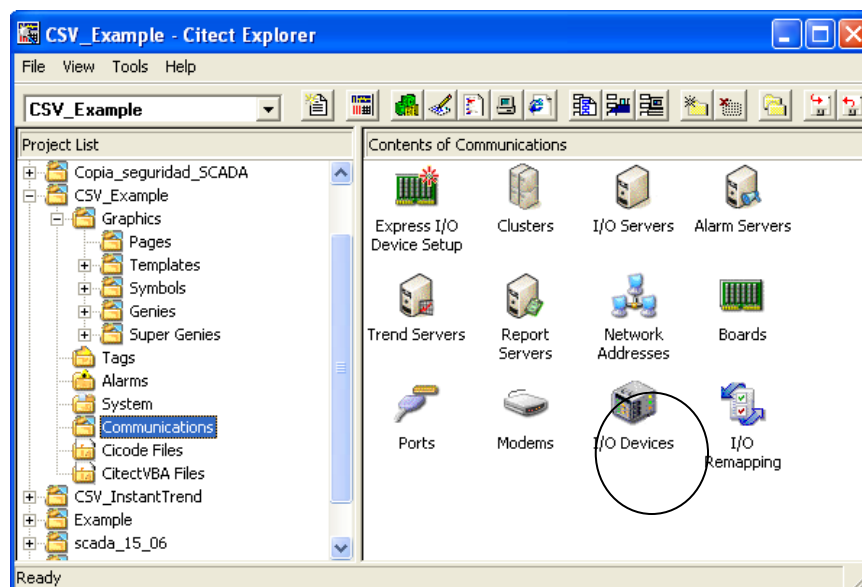


Ilustración 52: Apartado Comunicaciones en el explorador de proyectos.

Una vez allí, se añade el nuevo dispositivo a conectar, indicando ahora como protocolo MODNET (en vez de MODNET30, el protocolo que viene predeterminado). El nombre del puerto se cambia también a PORT2_BOARD2. Por último, se pulsa en *Reemplazar* para sobrescribir los cambios.

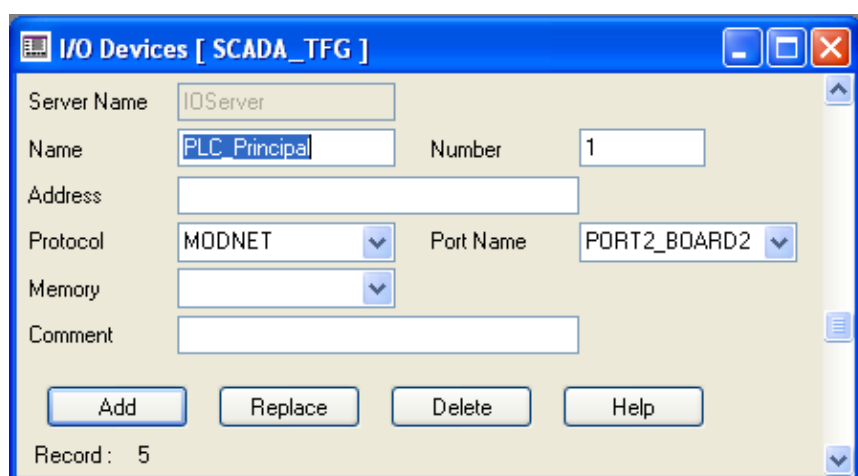


Ilustración 53 Modificación de la configuración del dispositivo de E/S.

Por último, se procede a importar las variables usando Unity SpeedLink Static. Para ello, en el explorador de proyectos, se pulsa en *Importar tags*, en el menú desplegable *Herramientas*.

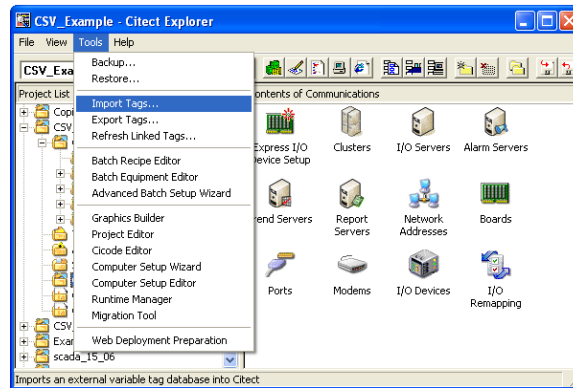


Ilustración 54: Importación de tags desde el explorador de proyectos.

En la ventana de importación, se selecciona el dispositivo de E/S configurado anteriormente. En el tipo de base de datos se selecciona Unity Speedlink Static. También es necesario poner el directorio en el que se encuentran las variables exportadas. Como opciones extras se encuentran el añadir un prefijo en la exportación de los tags, o el eliminado de los tags de este dispositivo de E/S antes de importar de nuevo.

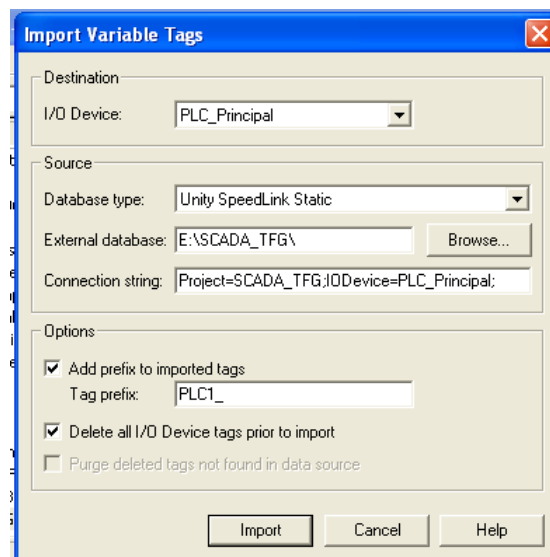


Ilustración 55: Ventana de exportación de variables.

Por último, se pulsa en *Importar* para proceder con la operación.

6.3. Uso de vectores

Es posible el uso de vectores usando comunicación Unity Speedlink static. Como ejemplo práctico, se tomará el paso de un vector de 7 componentes de variables tipo INT.

En primer lugar, se necesita que el vector se encuentre direccionado físicamente en el autómata. Para ello, en el *Editor de variables* se direcciona la cabecera del vector (en este caso se ha ubicado en la dirección %MW0). Es también necesario tener en cuenta el tipo de variable que se está direccionando: para el caso de tipo EBOOL se usa una dirección tipo %M, mientras para el resto de variables de debe usar una dirección tipo '%MW'.

Una vez direccionado, se creará otra variable que indicará el comienzo del vector en la memoria del autómata, la cual será usada posteriormente en el SCADA (en el ejemplo, corresponde con *vector_SCADA*). Se direcciona en la misma dirección que el primer elemento del vector.

vector	vector	%Mw0		
vector[0]	INT	%Mw0		
vector[1]	INT	%Mw1		
vector[2]	INT	%Mw2		
vector[3]	INT	%Mw3		
vector[4]	INT	%Mw4		
vector[5]	INT	%Mw5		
vector[6]	INT	%Mw6		
vector_SCADA	INT	%Mw0		

Ilustración 56: Direccionado del vector y su cabecera en Unity Pro XL.

Se procede a compilar el proyecto, y se darán 2 avisos en la compilación, indicando que dos variables se encuentran solapadas en la memoria del autómat.

Una vez importado los tags, se procede a la creación del vector en Vijeo Citect. Para ello, en la ventana de exploración de los tags de variables, se indica en el cuadro de texto de la dirección la dimensión del vector mediante el uso de corchetes (como se indica en la Ilustración 57: configuración del vector).

Ilustración 57: configuración del vector en Vijeo Citect.

Es necesario tener en cuenta en el uso de vectores en Vijeo Citect que la primera componente del vector se considera la componente 0.

6.4. Tabla comparativa entre Unity Speedlink y OPC

	Unity Speedlink	OPC
Conexión entre equipos del mismo fabricante	Sólo entre equipos de Schneider Electric.	Sí
Conexión entre equipos de distintos fabricantes	No	Sí
Posibilidad de tipos derivados	Vectores	No
Necesaria direccionamiento físico en memoria de variables	Sí	No
Uso de variables direccionadas en módulos de entrada o salida	No (existe procedimiento auxiliar)	Lectura/Escritura.

Ha de matizarse que sólo es posible la lectura o escritura de variables direccionadas en los módulos de entrada o salida usando Unity Speedlink mediante el uso de variables auxiliares. Para ello, se hará una asignación

directa de estas variables de salida con las variables situadas en los módulos de entrada y salida.

7 APLICACIÓN PRÁCTICA: CÉLULA DE FABRICACIÓN FLEXIBLE

Una vez exploradas todas las conexiones entre los distintos equipos, se presenta el ejemplo práctico de demostración, que incluye todas las conexiones presentadas en los capítulos anteriores.

En la aplicación de demostración, el objetivo será tomar un número determinado de pallets que se encuentren almacenados en el sistema de almacenamiento robótico, y procesarlos en el puesto del robot SCARA situado en el tramo 3 del sistema de transporte (por ejemplo, situando una pieza sobre el pallet). Una vez procesado el contenido del pallet, se almacenará de nuevo en el espacio definido para ellos en el sistema robótico.

Como se ha explicado en la descripción de la instalación realizada en el capítulo 3, actualmente el robot SCARA se encuentra fuera de servicio. Por ello, el procesamiento del pallet en este puesto (servir o retirar pieza del pallet) se simulará con una parada de 3 segundos en el puesto de trabajo del robot.

La aplicación consta de dos opciones en su funcionamiento:

- **Modo servir/retirar pieza:** Permite elegir la acción a realizar en el robot SCARA. En el caso de servir pieza, se extrae del sistema de almacenamiento de pallets un pallet no procesado; por el contrario, si se desea retirar pieza, el pallet debe estar ya procesado.
- **Modo automático:** Asigna las peticiones del número de piezas a procesar en función del estado actual del almacén. La elección del modo servir pieza y retirar pieza se realiza de forma automática cuando este modo esté activado.

7.1. Sistema de transporte

En este apartado se hace una descripción más detallada del sistema de transporte (sensores y actuadores controlados por el autómata principal) y del panel de control. También se presentará la guía GEMMA de la aplicación de demostración, así como de su implementación en el autómata usando Unity Pro XL.

Para la descripción del sistema de transporte, se ha tomado como referencia el trabajo de fin de grado [13]

7.1.1 Descripción del sistema de transporte

7.1.1.1 Descripción de los actuadores y sensores

Los actuadores y sensores se encuentran señalizados como cuadrados y círculos respectivamente.

S13	Sensor de bandeja en desvío 5.
S14	Sensor de bandeja en entrada al almacén.
S15	Sensor de bandeja en entrada al puesto de trabajo del SCARA.
S16	Sensor de bandeja en salida al puesto de trabajo del SCARA.
S17	Sensor de bandeja en entrada al punto previo del alimentador de bandejas.
S18	Sensor de bandeja en salida al punto previo del alimentador de bandejas.
S19	Sensor de final de tramo 5.



Ilustración 59: Sensores inductivos ubicados en el final del tramo.

Los sensores que se encuentran en el sistema de transporte son de tipo inductivo. Todos ellos (exceptuando los sensores S5 y S10) se encuentran fijados sobre la cinta en la parte interior, de forma que son activados por las bandejas (éstas poseen unas pequeñas chapas metálicas para su activación). Los sensores S5 y S10 se encuentran montados en el final de los tramos 2 y 4 como se muestra en la Ilustración 59. **¡Error! No se encuentra el origen de la referencia.** Estos sensores se encuentran siempre activados hasta que una bandeja presiona el tope del tramo correspondiente, produciendo su desactivación.

Tabla 24: Actuadores de la cinta transportadora.

Actuador (Y)	Descripción
Y0	Válvula general de aire de la instalación.
Y1	Retenedor previo al desvío 1.
Y2	Elevador del desvío 1.
Y3	Retenedor previo al desvío 2.
Y4	Elevador del desvío 2.
Y5	Retenedor previo al desvío 3.
Y6	Elevador del desvío 3.
Y7	Retenedor previo al desvío 4.

Y8	Elevador del desvío 4.
Y9	Retenedor previo al desvío 5.
Y10	Elevador del desvío 5.
Y11	Retenedor para el puesto de trabajo del almacén matricial.
Y12	Retenedor para el puesto de trabajo del robot SCARA.
Y13	Retenedor en el punto previo al alimentador de bandejas.
Y14	Retenedor del fin del tramo 5.

Los retenedores de bandejas son de tipo neumático, encontrándose fijados en las cintas transportadoras. En su estado de reposo se encuentran elevados.

Los elevadores de las intersecciones entre los diferentes tramos del sistema de transporte son también de tipo neumático. Normalmente se encuentran en su posición inferior, mientras que si se activan se elevan a su posición superior.



Ilustración 60: Retenedores (actuadores) del sistema de transporte.

7.1.1.2 Descripción de cuadros y panel de control

El panel de control se encuentra en la zona central de la instalación (en el interior de los distintos tramos del sistema de transporte). Presenta distintas zonas diferenciadas:

7.1.1.2.1 Indicadores luminosos

Indican el estado de los equipos auxiliares.

- Emergencia: El pulsador de emergencia del propio panel está activado.
- Parada motores: Se ha realizado una parada a los motores.

- Aire: La planta tiene presión suficiente para su funcionamiento.
- Tensión maniobra: Existe tensión en la instalación.

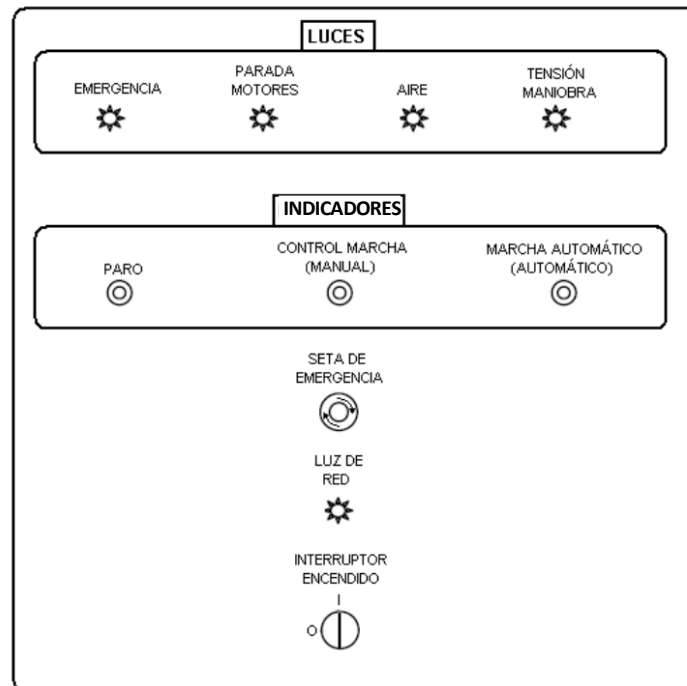


Ilustración 61: Panel de control de la instalación.

7.1.1.2.2 Pulsadores

Existen 3 pulsadores que permiten el cambio de un estado a otro en la guía GEMMA.

- Paro: Normalmente abierto. Permite realizar una petición de parada a la planta.
- Control marcha (indicado en el panel como *MODO BÚSQUEDA*): Normalmente abierto. Usado en la instalación para activar la puesta en marcha de la instalación.
- Marcha automático (indicado en el panel como *MODO AUTOMÁTICO*): Normalmente cerrado. Usado en la instalación para iniciar el modo de producción tras una puesta en marcha, o para reanudar la marcha tras haber solicitado un paro en la instalación.

Los dos últimos poseen un indicador luminoso que son activados cuando la acción está disponible.

7.1.1.2.3 Seta de emergencia

Situado en la parte central del panel de control. Produce la desconexión de todos los equipos de la instalación. Esta seta de emergencia no se encuentra conectada a ningún relé de seguridad, por lo que no es posible la implantación de una guía GEMMA que cuente con un estado de parada de emergencia que se active con la pulsación de ésta.

7.1.1.2.4 Interruptor de encendido

Conecta a la red eléctrica toda la instalación asociada al PLC principal (cintas de transporte y sistema de clasificación).

7.1.2 Guía GEMMA

7.1.2.1 Introducción a la guía GEMMA

En el estado de operación de la instalación, ésta no se encontrará de forma permanente en el estado de producción; existen distintos sucesos que obliga a la instalación a cambiar su estado. Un ejemplo de esto puede ser la aparición de un fallo grave, que obligue a detener la instalación, o un fallo leve en alguno de los sensores, que obligue a reducir la velocidad de la producción.

Con el fin de contemplar toda la casuística posible en la programación se han seguido los principios de la guía GEMMA⁴, desarrollada por la ADEPA⁵ a principios de la década de los 90. La guía GEMMA es una guía gráfica que muestra de forma clara los distintos estados posibles en los que puede estar nuestro proceso de producción, así como las condiciones necesarias para pasar de un estado a otro. Esta guía se encuentra recogida por la AFNOR (Agencia Francesa de Normalización) en la norma NF C03-1904.

El automatismo se compone de dos partes básicas: el sistema de producción y el control del mismo. El control puede estar a su vez en dos estados, con alimentación o sin ella.

Cuando el control se encuentra alimentado, en la guía GEMMA existen tres módulos diferenciados que aparecen representados como bloques: módulo de seguridad, módulo de modos de marcha, y módulo de producción. Además de estos bloques, también aparece representado otro rectángulo en línea discontinua que representa la parte operativa del proceso.

En la guía GEMMA se encuentran recogidos 17 posibles estados, que representan cada uno de los estados posibles en la máquina. En la Ilustración 62 se puede ver la guía GEMMA completa.

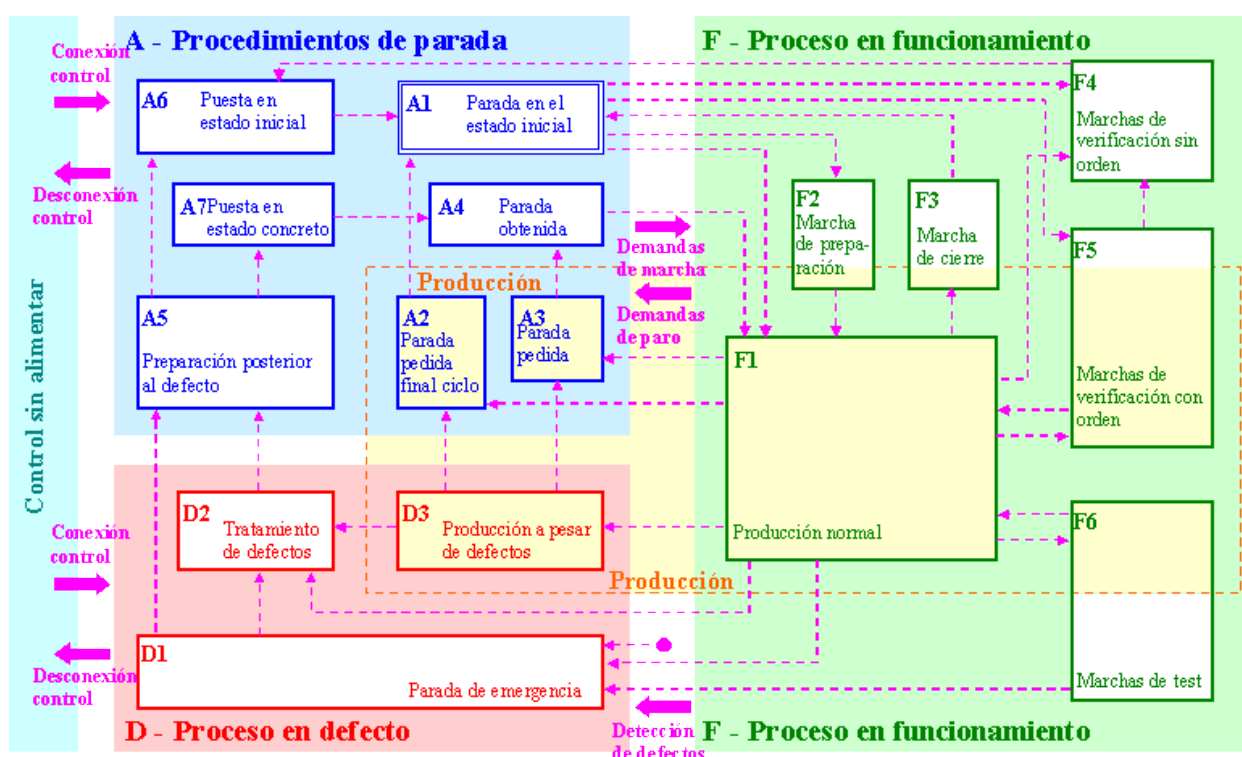


Ilustración 62: Guía GEMMA [14].

7.1.2.2 Presentación de la guía GEMMA implementada

7.1.2.2.1 A6 – Puesta en estado inicial

Es el primer estado por el que pasa la instalación tras el encendido. Para acceder a él, se pulsará el pulsador MODO BÚSQUEDA (se encontrará iluminado en verde cuando esté disponible la acción).

En este estado, primero se activarán los motores y los tramos de desvío, y se irán acumulando las bandejas disponibles en el sistema de transporte en los tramos 2 y 4. La desconexión de los motores de los tramos en la acumulación de bandejas se realizará de forma progresiva, conforme las bandejas vayan desplazándose hacia los tramos 2 y 4. Una vez finalizada la acumulación, los retenedores del final de los tramos irá permitiendo el paso de las bandejas de forma individual con el fin de contar el número total. Una vez conocido el número de bandejas actual en la instalación, se igualará el número de bandejas en el sistema de transporte al número de

⁴ Guide d'Études des Modes de Marches et de Arrêts.

⁵ Agence nationale pour le Développement de la Production Appliquée à l'industrie.

peticiones de almacenamiento al alimentador de bandejas.

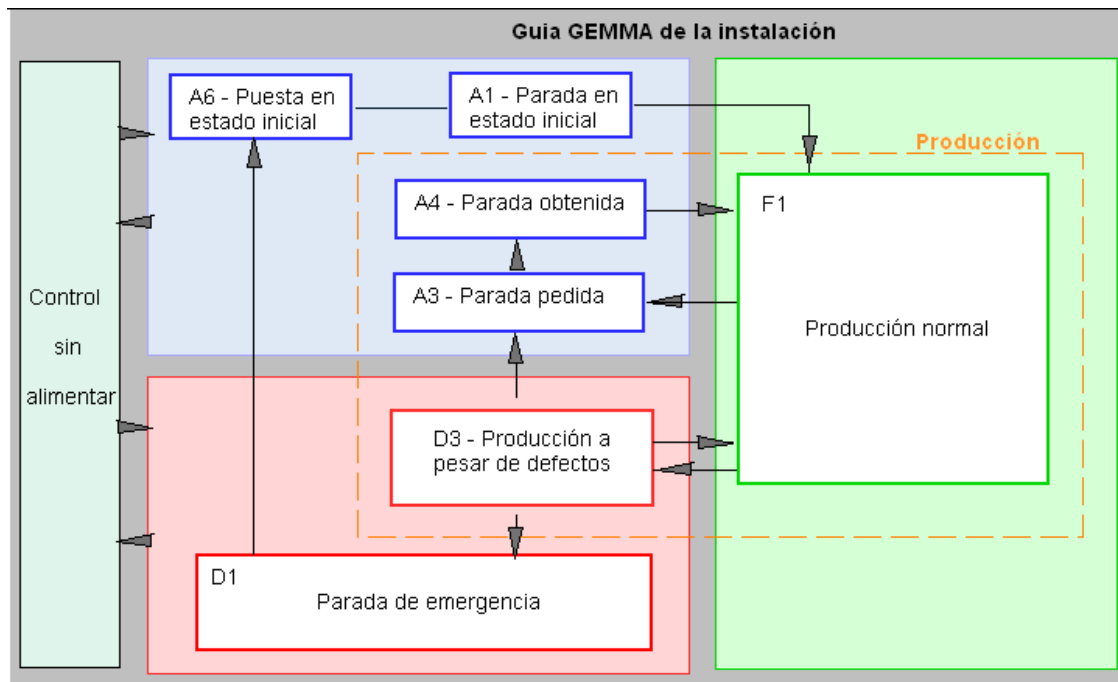


Ilustración 63: Guía GEMMA implementada en la instalación.

En el caso de que la bandeja que se vaya a almacenar tenga un pallet sobre ella, será considerado como pallet defectuoso, almacenándose en la zona de pallets defectuosos en el sistema robotizado de almacenaje.

7.1.2.2.2 A1- Parada en estado inicial

Una vez realizada la puesta en marcha por parte del almacén, permanecerá en el estado inicial hasta que el pulsador del panel indicado como *MODO AUTOMÁTICO* (estará iluminado en azul) sea pulsado. Este pulsador como su nombre indica dará paso al modo automático, es decir, al estado de *F1- Producción Normal*.

7.1.2.2.3 F1-Producción normal

Estado también conocido como modo automático. Se accederá a este estado cuando el pulsador del panel indicado como *MODO AUTOMÁTICO* (estará iluminado en azul) sea pulsado. En este estado se lleva a cabo el procesamiento de las distintas piezas. Se distinguen tres modos de funcionamiento (como se ha indicado al principio del capítulo):

- **Modo servir pieza:** Se sirven pallets del sistema robotizado de almacenaje que se encuentren sin procesar (es decir, sin pieza), se llevan al puesto del robot SCARA en la que permanecen 3 segundos (para simular el procedimiento de servir pieza), y se almacenarán de nuevo como pallets procesados (es decir, con piezas).
- **Modo retirar pieza:** Realiza el ciclo inverso. Se sirven pallets del sistema robotizado de almacenaje que se encuentren procesados (es decir, con pieza), se llevan al puesto del robot SCARA en la que permanecen 3 segundos (para simular el procedimiento de retirar pieza), y se almacenan de nuevo como pallets sin procesar.
- **Modo de peticiones automático:** Modo predeterminado de funcionamiento. El autómata asigna las peticiones de retirar o servir piezas en función del estado actual de ocupación del sistema robotizado de almacenaje (número de estantes ocupados y vacantes de cada categoría).

Las peticiones son limitadas en función de los huecos y piezas disponibles en el almacén en todos los casos. Esto quiere decir que, aunque la petición que se le encargue a la instalación supere a las piezas disponibles, esta se limitará a las posibilidades del almacén en ese momento.

Las bandejas son servidas o almacenadas por el alimentador de las bandejas en función de las peticiones pendientes y peticiones en proceso. El número de bandejas máximo en la instalación es configurable, con un límite de 6 bandejas circulando a la vez en la instalación.

7.1.2.2.4 A3- Parada pedida

En este estado se lleva a cabo el procedimiento para realizar la parada de la planta. El procedimiento consiste en terminar las peticiones que se estén llevando a cabo en ese momento en la instalación, y guardar en memoria aquellas que se han pospuesto.

Para acceder a él, se pulsará el pulsador de paro situado en el panel de control de la instalación.

7.1.2.2.5 A4- Parada obtenida

Estado alcanzado tras realizar una parada durante el estado de producción normal (F1). Para reanudar la marcha y regresar al estado de producción normal, se debe pulsar en MODO AUTO (tendrá el indicador luminoso encendido).

7.1.2.2.6 D3-Producción a pesar de defectos

Estado alcanzado cuando se produce un defecto leve. Los defectos leves son aquellos que permiten el funcionamiento de la planta, aunque con capacidad limitada. Los defectos leves considerados son el fallo de algún sensor de alguno de los desvíos, o la existencia de un número de bandejas en el alimentador menor al número de bandejas máximo en la instalación.

7.1.2.2.7 D1- Parada de emergencia

Estado alcanzado en el caso de producirse algún error grave (error de comunicación con el almacén, error grave en el alimentador de bandejas, pérdida de bandeja en el sistema de transporte, ...).

Para abandonar el estado de parada de emergencia se debe solucionar el error, y resetear el error grave de la instalación. Tras este estado, el estado de la instalación evolucionará hacia una puesta en marcha.

7.1.3 Implementación usando Unity Pro XL

7.1.3.1 Descripción del PLC

La configuración del autómatas usado es el siguiente:

- Rack BMX XBP 0800. Admite hasta un máximo de 7 módulos de entradas, salidas o de funciones especiales.
- Módulo de alimentación de red de corriente alterna BMX CPS 2000. Proporciona una corriente continua de 24V.
- Módulo procesador BMX P34 2020, con conexión puerto serie, Ethernet y USB (explicado previamente en los capítulos de conexión).
- Módulo de entradas digitales BMX DDI 6402K de lógica positiva con canales de entrada que reciben la corriente de los sensores.
- Módulo de salidas digitales BMX DDO 6402K de lógica positiva con canales de salida que proporcionan corriente a los actuadores.

7.1.3.2 Configuración del PLC usando Unity Pro XL

Se deben indicar los módulos existentes en el autómatas que se desea programar. Para ello, en el explorador de proyectos se abre la carpeta *Configuración*, y se seleccionan cada uno de los módulos.

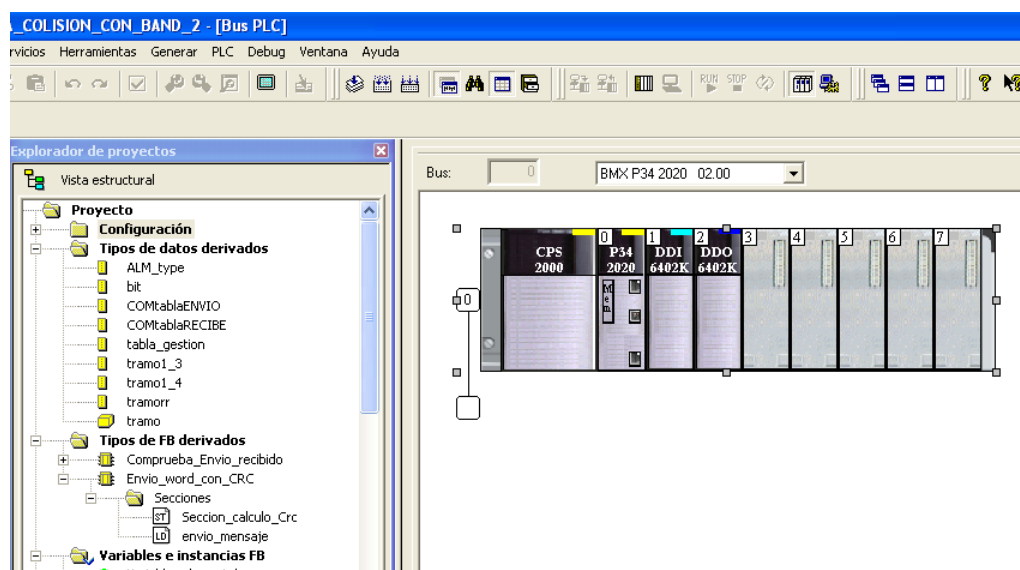


Ilustración 64: Configuración de los módulos disponibles en el autómata.

Para la configuración de las conexiones, es necesario seguir los pasos indicados en la sección 5.2, donde se explica el procedimiento para la conexión a la red Ethernet.

Una vez configurada la red Ethernet, es posible realizar la conexión del PC al autómata mediante ella. Para proceder a la conexión, se accederá a *Establecer conexión*, situado en el menú desplegable *PLC*.

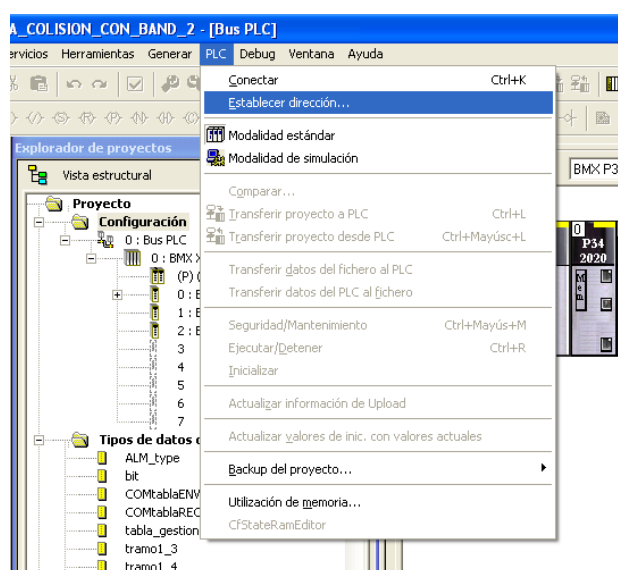


Ilustración 65: Acceso a la ventana de conexión al autómata.

En ella, se configuran los parámetros de conexión al autómata deseado (dirección IP y medios). En el caso de autómata principal, la *dirección IP* es 192.168.0.12 y en el campo *Medios* se seleccionará TCP/IP. Tras la configuración, es necesario que se pulse en aceptar y se cierre la ventana con el fin de validar la configuración.

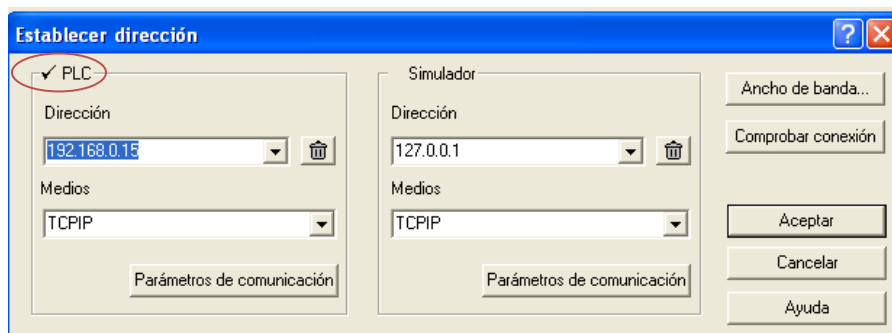


Ilustración 66: Cuadro para establecer conexión con el autómata usando la red Ethernet.

También se puede realizar la conexión mediante USB. Para ello, en la ventana emergente se selecciona en la categoría *Medios* la opción USB, y en la dirección se indica SYS. Para comprobar que los parámetros son correctos se pulsa en *Comprobar conexión*; en el caso de conexión correcta, se pulsa aceptar para validar los cambios.

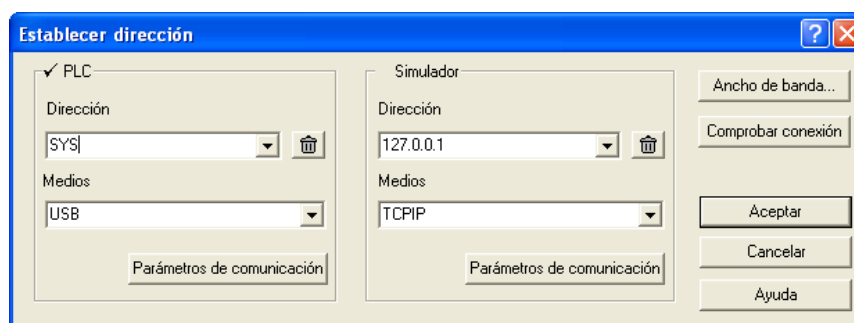


Ilustración 67: Ventana emergente para configurar la dirección.

7.1.3.3 Direcciones de entradas y salidas al autómata

7.1.3.3.1 Direcciones de entrada

Tabla 25: Entradas del autómata principal.

Nombre variable	Dirección	Descripción
ED_PRESOSTATO	%I0.1.0	Sensor de presión de aire de la instalación.
ED_ENT_DESVIO_1	%I0.1.1	Sensor de bandeja en entrada al desvío 1.
ED_PALET_ELEV_1	%I0.1.2	Sensor de bandeja en desvío 1.
ED_SAL_DESVIO_1	%I0.1.3	Sensor de bandeja en salida del desvío 1.
ED_ENT_DESVIO_2	%I0.1.4	Sensor de bandeja en entrada al desvío 2.
ED_PALET_ELEV_PUSH_2	%I0.1.5	Sensor de bandeja en desvío 2 (final de tramo)
ED_ENT_DESVIO_3	%I0.1.6	Sensor de bandeja en entrada al desvío 3.
ED_PALET_ELEV_3	%I0.1.7	Sensor de bandeja en desvío 3.
ED_SAL_DESVIO_3	%I0.1.8	Sensor de bandeja en salida del desvío 3.
ED_ENT_DESVIO_4	%I0.1.9	Sensor de bandeja en entrada al desvío 4.

ED_PALET_ELEV_PUSH_4	%I0.1.10	Sensor de bandeja en desvío 4 (final de tramo).
ED_ENT_DESVIO_5	%I0.1.11	Sensor de bandeja en entrada al desvío 5.
ED_PALET_ELEV_5	%I0.1.12	Sensor de bandeja en desvío 5.
ED_PALET_ELEV_PUSH_5	%I0.1.13	Sensor de bandeja en desvío 5 (final de tramo).
ED_ENT_ALMACEN	%I0.1.14	Sensor de bandeja en entrada al almacén.
ED_ENT_SCARA	%I0.1.15	Sensor de bandeja en entrada al robot SCARA.
ED_SAL_SCARA	%I0.1.16	Sensor de bandeja en salida del robot SCARA.
ED_ENT_DISP_BANDEJA	%I0.1.17	Sensor de bandeja en entrada al alimentador de bandejas.
ED_SAL_DISP_BANDEJA	%I0.1.18	Sensor de bandeja en salida al alimentador de bandejas.
ED_PUL_BUSQUEDA	%I0.1.21	Pulsador de búsqueda del panel de mandos.
ED_PUL_AUTOMATICO	%I0.1.22	Pulsador de marcha automático del panel de mandos.
ED_PUL_PARADA	%I0.1.23	Pulsador de parada del panel de mandos.

7.1.3.3.2 Direcciones de salida

Tabla 26: Salidas del autómatas principal.

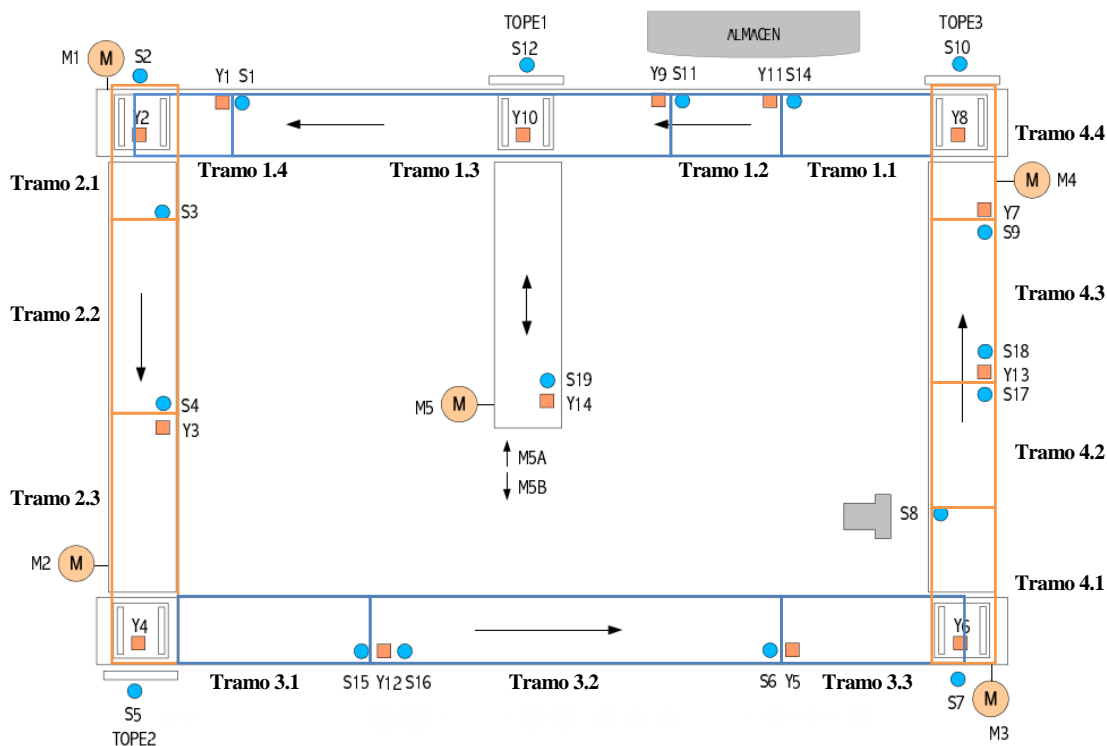
Nombre de la variable	Dirección	Descripción
SD_VAL_GENERAL	%Q0.2.0	Válvula general.
SD_RET_ENT_DESVIO_1	%Q0.2.1	Descenso del retenedor previo al desvío 1.
SD_ELEV_DESVIO_1	%Q0.2.2	Elevación del desvío 1.
SD_RET_ENT_DESVIO_2	%Q0.2.3	Descenso del retenedor previo al desvío 2.
SD_ELEV_DESVIO_2	%Q0.2.4	Elevación del desvío 2.
SD_RET_ENT_DESVIO_3	%Q0.2.5	Descenso del retenedor previo al desvío 3.
SD_ELEV_DESVIO_3	%Q0.2.6	Elevación del desvío 3.
SD_RET_ENT_DESVIO_4	%Q0.2.7	Descenso del retenedor previo al desvío 4.
SD_ELEV_DESVIO_4	%Q0.2.8	Elevación del desvío 4.
SD_RET_ENT_DESVIO_5	%Q0.2.11	Descenso del retenedor previo al desvío 5.
SD_RET_ALMACEN	%Q0.2.12	Descenso del retenedor previo al almacén.
SD_RET_DISP_BANDEJA	%Q0.2.13	Descenso del retenedor previo al alimentador de bandejas.

SD_MOTOR_1	%Q0.2.16	Motor del tramo 1.
SD_MOTOR_2	%Q0.2.17	Motor del tramo 2.
SD_MOTOR_3	%Q0.2.18	Motor del tramo 3.
SD_MOTOR_4	%Q0.2.19	Motor del tramo 4.
SD_PILOTO_MODO_BUSQUEDA	%Q0.2.22	Piloto del pulsador de MODO BÚSQUEDA.
SD_PILOTO_PARADA_MOTORES	%Q0.2.23	Piloto del panel de control PARADA MOTORES.
SD_PILOTO_MODO_AUTO	%Q0.2.24	Piloto del panel de control MARCHA AUTOMATICO.

7.1.3.4 Programación

En primer lugar, con el fin de la comprensión de la programación realizada, se explica cómo se han gestionado los datos de cada una de las bandejas que se encuentran circulando en el modo de producción.

Se ha dividido cada tramo del circuito del sistema de transporte en 14 sub-tramos (los tramos 1 y 4 se han dividido en 4 sub-tramos, mientras que los tramos 2 y 3 se han dividido en 3 sub-tramos).



Cada sub-tramo lleva asociada una estructura de tipo Tramo, que ha sido definida como una variable derivada. Consta de los siguientes campos:

- Bandeja (EBOOL): Indica si hay bandeja en el sub-tramo.
- Pallet (EBOOL): Indica si la bandeja lleva un pallet.
- Procesado (EBOOL): Indica si el pallet que lleva está procesado.
- Tiempo (INT): Número de pulsos de 125ms que se han dado desde que entró la bandeja al sub-tramo.
- Defecto (EBOOL): Indica con un 1 si la bandeja está defectuosa. No se usa este campo

en la aplicación actual.

A su vez, cada tramo, lleva asociado un vector de estructuras Tramo. Se han definido dos tipos de datos derivados para los tramos de 3 y 4 sub-tramos:

- Tramo_3: Vector de tres componentes de tipo Tramo.
- Tramo_4: Vector de cuatro componentes de tipo Tramo.

Por último, se han declarado 4 variables:

Nombre de la variable	Tipo de dato	Descripción
Tramo1	Tramo_4	Variable asociada al tramo 1.
Tramo2	Tramo_3	Variable asociada al tramo 2.
Tramo3	Tramo_3	Variable asociada al tramo 3.
Tramo4	Tramo_4	Variable asociada al tramo 4.

En cada uno de los sub-tramos no puede haber dos bandejas al mismo tiempo, pues implicaría la pérdida de la información actual.

La programación se compone de un conjunto de 21 secciones. La programación completa puede consultarse en Anexo A. El objetivo de cada sección de programación será explicado a continuación en distintos apartados.

7.1.3.4.1 Valvula_General_y_Temp

Mantiene la válvula general de aire comprimido activada. En esta sección también se encuentran los temporizadores usados para los errores leves de desvío de los desvíos y el temporizador usado para simular el procesamiento de la pieza en el puesto de trabajo del robot SCARA.

7.1.3.4.2 Desvio_1 (SFC)

Se encarga de la gestión del desvío entre los tramos 1 y 2. Esta sección se encuentra condicionada con la variable *COND_DESVIO_1*.

En el caso de que no haya ningún fallo en los sensores del tramo, la secuencia seguida es la siguiente:

Se espera a la llegada de una bandeja al desvío. Cuando el sensor de entrada a la sección es activado (*ED_ENT_DESVIO_1* a nivel alto), se hace una espera de 500ms y se eleva el retenedor de bandeja previa del desvío (*ED_PALET_ELEV_1*). Una vez se detecte la llegada de la bandeja al final del tramo 1, si el tramo 2.1 se encuentra libre (es decir, si *Tramo2[1].bandeja* está a nivel bajo), se eleva el desvío (*SD_ELEV_DESVIO_1*) hasta que sea detectada la bandeja en la salida del desvío (*ED_SAL_DESVIO_1*). Por último, se procede a la bajada del retenedor de bandeja previa.

El caso de fallo de alguno de los sensores se encuentra también contemplado en esta sección, indicándose con la variable *FALLO_D1* a nivel alto.

7.1.3.4.3 Desvio_2 (SFC)

Se encarga de la gestión del desvío entre los tramos 2 y 3. Esta sección se encuentra condicionada con la variable *COND_DESVIO_2*.

En el caso de que no haya ningún fallo en los sensores del tramo, la secuencia seguida es la siguiente:

Se espera a la llegada de una bandeja al desvío con el desvío elevado. Cuando el sensor de entrada al desvío es activado (*ED_ENT_DESVIO_2* a nivel alto), se hace una espera de 500ms y se eleva el retenedor de bandeja previa del desvío. Una vez se detecte la llegada de la bandeja a final del tramo (*SD_ELEV_DESVIO_PUSH_2* a nivel bajo), si el tramo 3.1 se encuentra libre (*Tramo3[1].bandeja* a nivel bajo) se procede a poner el elevador en su posición inferior. Por último, cuando el sensor de bandeja tipo WIPER vuelva a nivel alto (*ED_ELEV_DESVIO_PUSH_2*), el desvío se vuelve a poner en su posición elevada, y el retenedor de

bandejas previa descende.

El fallo de todos los sensores está contemplado (al igual que en la sección del desvío 1), indicándose el fallo con la variable *FALLO_D2* a nivel alto.

7.1.3.4.4 Desvio_3 (SFC)

Esta sección es análoga a la sección *Desvio_1*, exceptuando la elevación del desvío (se encontrará a la entrada de la bandeja en el desvío en su posición inferior, y con la llegada de la bandeja al final del tramo pasará a su posición superior). También se encuentra condicionada con la variable *COND_DESVIO_3*.

7.1.3.4.5 Desvio_4 (SFC)

Esta sección es análoga a la sección *Desvio_2*, exceptuando la elevación del desvío (se encontrará a la entrada de la bandeja en su posición inferior, y con la llegada de la bandeja al final del tramo pasará a su posición superior). También se encuentra condicionada con la variable *COND_DESVIO_4*.

7.1.3.4.6 ACUMULACION (LD)

Se encarga de chequear durante la puesta en marcha cuándo se puede proceder al apagado de los motores que mueven cada uno de los tramos del sistema de transporte. Esta sección funcionará sólo en el caso de que la variable *INICIO_ACUMULACION* se encuentre activada.

En cada uno de los tramos, se espera a que pasen 10 segundos sin que ningún sensor de los que se encuentran en el tramo correspondiente sea activado. En el caso de cumplir esta condición, se activará la variable *FIN_ACUMULACION_n*, siendo *n* el número del tramo correspondiente.

7.1.3.4.7 CONTADOR_VEH (LD)

En esta sección se encuentran implementados los contadores que llevan a cabo la cuenta del número de vehículos de los tramos 1 y 3.

Se ha usado un contador *CTU_INT* (contadores de cuenta positiva de tipo entero) para cada uno de los tramos. Cuando la variable *INICIO_CUENTA_VEH* se active, reseteará la cuenta actual en los contadores y comenzará a contar el número de flancos de bajada que se produzcan en el sensor de entrada del desvío (*ED_ENT_DESVIO_2* o *ED_ENT_DESVIO_4*), es decir, vehículos que entran a los tramos 1 y 3.

7.1.3.4.8 Puesta_marcha_SFC (SFC)

En esta sección se implementa el estado de puesta en estado inicial (A6) descrito en la guía GEMMA.

En primer lugar, se desactivan las secciones *DESVIO_1*, *DESVIO_2*, *DESVIO_3* y *DESVIO_4*, se elevan los retenedores de los desvíos y se hace descender a los retenedores intermedios de los tramos. Tras esto, se elevan los desvíos durante 2 segundos para vaciar las posibles bandejas que hayan quedado en el desvío fuera del alcance de detección de los sensores, y se procede a la activación de las secciones de desvíos y a la variable bandera que indica el inicio de la acumulación (realizada por la sección Ladder *ACUMULACION*).

Conforme las variables *FIN_ACUMULACION_n* vayan siendo activadas, se irá realizando el apagado del motor del tramo *n* correspondiente. Tras finalizar la acumulación, se hacen descender los retenedores de las secciones y se activa la cuenta de las bandejas durante 20 segundos. Transcurrido ese tiempo, se realiza la escritura de las peticiones de almacenamiento remotas al alimentador de piezas.

En el caso de que se detecte una bandeja con pallet, entonces se eleva el retenedor del puesto de trabajo del almacén. Una vez que la pieza sea detectada por el sensor, se solicita el almacenamiento de un pallet defectuoso. Cuando el sistema robotizado de almacenaje haya recogido el pallet de la bandeja, se hace descender el retenedor, dejando avanzar a la bandeja hacia el alimentador de bandejas.

Una vez que el número de peticiones remotas de almacenamiento en el alimentador sea 0, se procede a activar una variable bandera indicando el final de la puesta en marcha.

7.1.3.4.9 MODO_AUTO (SFC)

Esta sección implementa los estados A4 (Parada obtenida), A3 (Parada pedida), F1 (Producción normal) y D3 (Producción a pesar de defectos) explicados en la guía GEMMA. Se encuentra condicionada con la variable *COND_MODO_AUTO*.

En primer lugar, se inicia con la pulsación del pulsador *MODO AUTOMÁTICO*. Con esto, se activarán los motores y las secciones que rigen los desvíos. En el caso de una parada solicitada, esperará la finalización de la acción de servir o almacenar bandeja o pallet (en el caso de que se esté realizando en ese momento) y guarda el número de peticiones pendientes en una variable llamada *Pet_preparo*. Una vez guardadas, se establece el número de peticiones al número de peticiones en proceso y se espera a que termine de realizarlas. Por último, se apagan los desvíos y los motores de todos los tramos.

7.1.3.4.10 Comunicación_MODBUS (LD)

Está compuesta por los distintos bloques relacionados con la comunicación Modbus entre el autómatas principal y el alimentador de bandejas.

En ella se encuentran un bloque ADDM, dos bloques READ_VAR para la lectura de los tipos EBOOL e INT del autómatas del alimentador de bandejas, un bloque WRITE_VAR para la escritura de las peticiones remotas, y los bloques INT_TO_WORD y WORD_TO_BIT con el fin de realizar la conversión de tipo INT a tipo BIT y poder acceder a los bits leídos.

7.1.3.4.11 ComunicacionPuertoSerie (LD)

Realiza la gestión de la comunicación por puerto serie. Para ello usa un bloque ADDM y los bloques creados para la gestión de la comunicación con CRC (*Lectura_mensaje_recibido_Con_CRC* y *Envio_mensaje_con_CRC*).

7.1.3.4.12 MODO_OPERA_ALMACEN (SFC)

Esta sección en SFC realiza la gestión de la comunicación mostrada en la Ilustración 12. Ilustración 12: Posibilidades en la comunicación del apartado 4.2.

7.1.3.4.13 Lee_matriz_sfc (SFC)

Esta sección se encarga de la recepción del vector de ocupación que el almacén envía cada vez que lleva a cabo una petición. La recepción se activa mediante un flanco de subida en la variable *inicio_recepcion_matriz*, e indica su final realizando un flanco de bajada en esta misma variable. Los distintos vectores recibidos son almacenados en el vector *Almacen* de 72 componentes.

En el caso de error de recepción de la matriz, activa la variable *Error_almacen_recepcion_mat* durante 2 segundos antes de realizar el flanco de bajada para indicar el final de la recepción.

7.1.3.4.14 Generador_pulsos (LD)

Contiene dos generadores de pulsos que serán usados en el resto de secciones. Uno de ellos genera pulsos cada 125ms (variable *salida_pulsos*), mientras que el otro genera pulsos cada 200ms (variable *salida_pulsos_200ms*).

7.1.3.4.15 SeguidorVisual (ST)

Actualiza toda la información contenida en las estructuras de cada tramo, es decir, la existencia de bandejas, pallets o piezas, y el tiempo de permanencia en cada tramo. Esta información se guarda en la estructura explicada al principio del apartado de programación. Para ello, en caso de activación de alguno de los sub-tramos, recupera la información del tramo anterior, y luego procede a su desactivación. El tiempo del tramo que se activa es reseteado antes de comenzar la cuenta del tiempo. En el caso de que este tiempo de permanencia de una bandeja supere un determinado límite en un sub-tramo, se considerará que la bandeja ha sido perdida.

También realiza en cada ciclo el cálculo del número de bandejas con pallets con piezas, con pallets sin piezas y bandejas libres circulando en la instalación.

7.1.3.4.16 ProgramacionRetenedoresAUTO (LD)

Realiza el control sobre el control de los distintos retenedores de los tramos. Esta sección se encuentra condicionada por una variable *COND_PROG_RET_AUTO*, que será activada cuando la instalación se encuentre en el modo de producción normal (F1).

7.1.3.4.17 Gestion_peticiones_bandejas (ST)

Gestiona las peticiones a los distintos puestos de trabajo (alimentador de piezas, robot SCARA y sistema robotizado de almacenaje). En esta sección se encuentra implementado también el modo de peticiones automáticas.

7.1.3.4.18 Supervision_errores (LD)

Supervisa la operación y detecta los fallos leves y graves que se produzcan durante la operación de la instalación.

El fallo grave de la instalación puede ser producido por:

- Número de bandejas incongruente: Se produce en caso de que el número de bandejas que se ha hallado mediante la información de cada tramo sea distinto al número de bandejas que se contabilizan cada vez que realiza el alimentador de bandejas una petición de servir o almacenar pieza. Este fallo se indica con la activación de la variable *Fallo_grave_bandejas*.
- Error de realización de petición en el sistema de almacenamiento: El error de realización se da cuando no ha sido posible realizar la petición debido al estado actual de la ocupación de los estantes. Se indica el error con la variable *error_realizacion*.
- Fallo grave del alimentador de bandejas: Es leída del autómatas del alimentador en la sección *ComunicacionModbus* a través de conexión Ethernet.
- Alimentador de bandejas en modo local: El alimentador de bandejas necesita encontrarse con el modo auto remoto activado en el control de mandos de la máquina. La variable que indica que se encuentra en modo remoto es leída en la sección *ComunicacionModbus* a través de la conexión Ethernet entre autómatas.
- Pérdida de bandeja en un tramo: Cada tramo tiene una limitación de tiempo máximo en la que la bandeja puede permanecer en él. En el caso de que este límite sea superado, se supone que la bandeja ha sido perdida en ese tramo (caída, pérdida, ...). Este fallo se indica con la componente correspondiente en el vector *fallo_grave_tramos*[14].

Para el reseteo de los fallos graves, se debe en primer lugar cambiar las condiciones que produjeron el fallo. Tras esto, se repara el fallo grave a través del sistema SCADA pulsando el pulsador indicado para ello, o pulsando los botones de *MODO AUTOMÁTICO* y *MODO BÚSQUEDA* en el cuadro de mandos de la instalación.

El fallo leve de la instalación puede ser producido por 2 motivos:

- Fallo leve en alguno de los desvíos: Puede ser producido por el fallo de uno de los sensores en el desvío o por la manipulación del desvío por parte de un operador (por ejemplo, activación de los sensores sin que se encuentre una bandeja).
- Nivel insuficiente de piezas en el alimentador: Se produce en el caso de que el alimentador se encuentre vacío y el número de bandejas en la instalación sea menor al número máximo de bandejas en la instalación.

7.1.3.4.19 Reseteos_tramos_tiempos (ST)

Resetea la información acerca de la existencia de bandejas o el tiempo de permanencia de éstas en cada uno de los tramos de la instalación. Este reseteo se encuentra condicionado por dos variables: *reparacion_fallo_grave* (variable que resetea los fallos graves de la instalación) o *reseteos_tramos* (variable que es usada en la guía GEMMA para resetear la instalación a su estado inicial).

7.1.3.4.20 Asignacion_variables_SCADA

Asigna las variables direccionadas en los módulos de entrada y salida del autómatas a variables direccionadas en la memoria del autómatas. Estas variables son indicadas con el prefijo *SC_*. También asigna a las estructuras de cada tramo variables para que puedan ser usadas en el sistema SCADA.

7.1.3.4.21 Reseteos_congelacion_secciones

En esta sección se lleva a cabo la inicialización de las secciones SFC en el caso de un reseteo completo de la instalación. Esto puede darse con el encendido del PLC o tras la salida del estado de emergencia.

También lleva a cabo la congelación de las secciones de los desvíos en el caso de encontrarse en una parada, con el objetivo de evitar que, aunque se activen los sensores de forma manual, su activación no sea detectada como un fallo leve del desvío.

7.2 Alimentador de bandejas

La comunicación entre los distintos puestos de trabajo en una célula de fabricación flexible se realiza usando bandejas que se moverán sobre unas cintas transportadoras. Estas bandejas podrán ser servidas o almacenadas por el alimentador de bandejas.

En este capítulo se hace una descripción física de la planta (autómata utilizado, sensores y actuadores) y se presenta la guía GEMMA implementada. Para el desarrollo de la descripción del alimentador se ha tomado como referencia la descripción realizada por Antonio Luis Velasco Martín en su trabajo de fin de grado: ‘Integración de alimentador de bandejas a célula de fabricación flexible controlada por PLC’ [15].

7.2.1 Descripción de la máquina.

7.2.1.1 Descripción de actuadores y sensores

El alimentador de bandejas consta de 4 cilindros y 11 sensores. En la Ilustración 68 se presenta un esquema de la instalación con los sensores y actuadores que se disponen.

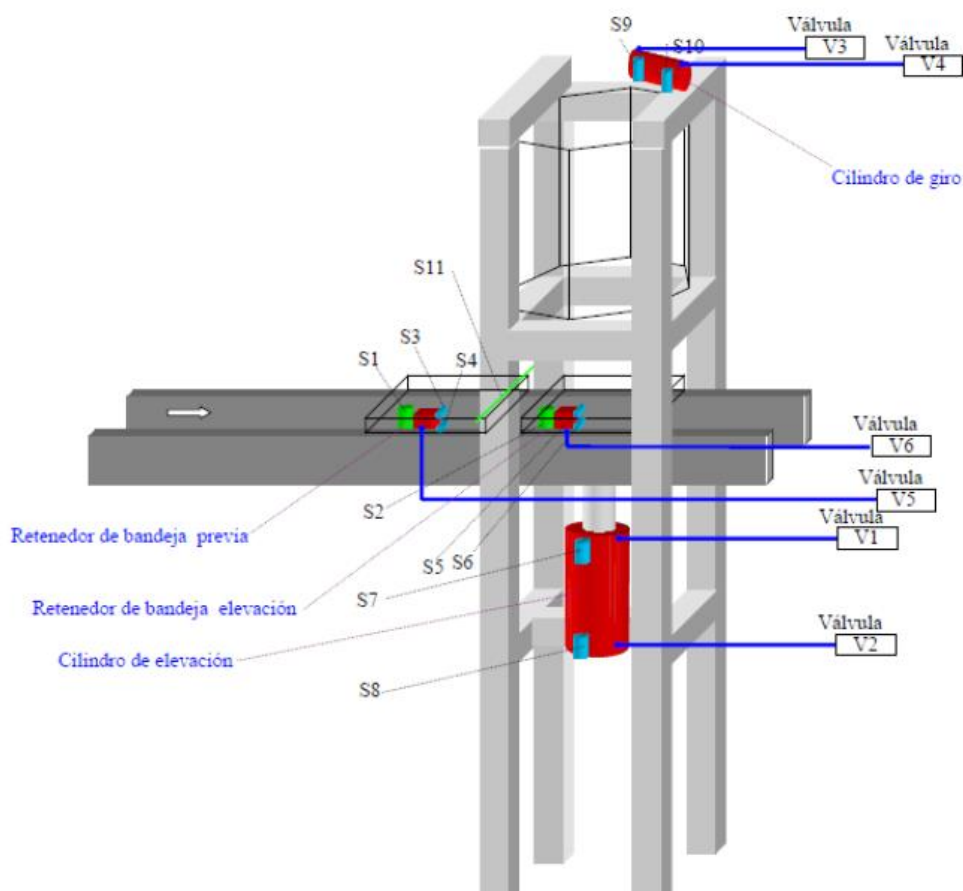


Ilustración 68: Esquema general de la distribución de sensores y actuadores.

Tabla 27: Descripción de los sensores.

Sensor	Descripción
S1	Sensor inductivo de bandeja en zona previa.
S2	Sensor inductivo de bandeja en zona de elevación.
S3	Sensor del retenedor de zona previa en posición elevada.
S4	Sensor del retenedor de zona previa en posición de reposo.
S5	Sensor del retenedor de zona de elevación en posición elevada.
S6	Sensor del retenedor de zona de elevación en posición de reposo.
S7	Sensor de cilindro de elevación en posición elevada.
S8	Sensor de cilindro de elevación en posición de reposo.
S9	Sensor izquierdo del cilindro de giro.
S10	Sensor derecho del cilindro de giro.
S11	Sensor de barrido para detectar bandejas u objetos.

Tabla 28: Descripción de los actuadores.

Actuador	Descripción
V1	Bajar cilindro de elevación.
V2	Subir cilindro de elevación.
V3	Rotación sentido horario con el cilindro de giro.
V4	Rotación sentido anti-horario con el cilindro de giro.
V5	Retenedor de bandeja en zona previa.
V6	Retenedor de bandeja en zona de elevación.

7.2.1.2 Descripción del panel de control

El alimentador de bandejas cuenta con un panel de control situado próximo al tramo 3 de las cintas transportadoras. Este panel cuenta con 8 secciones diferenciadas:

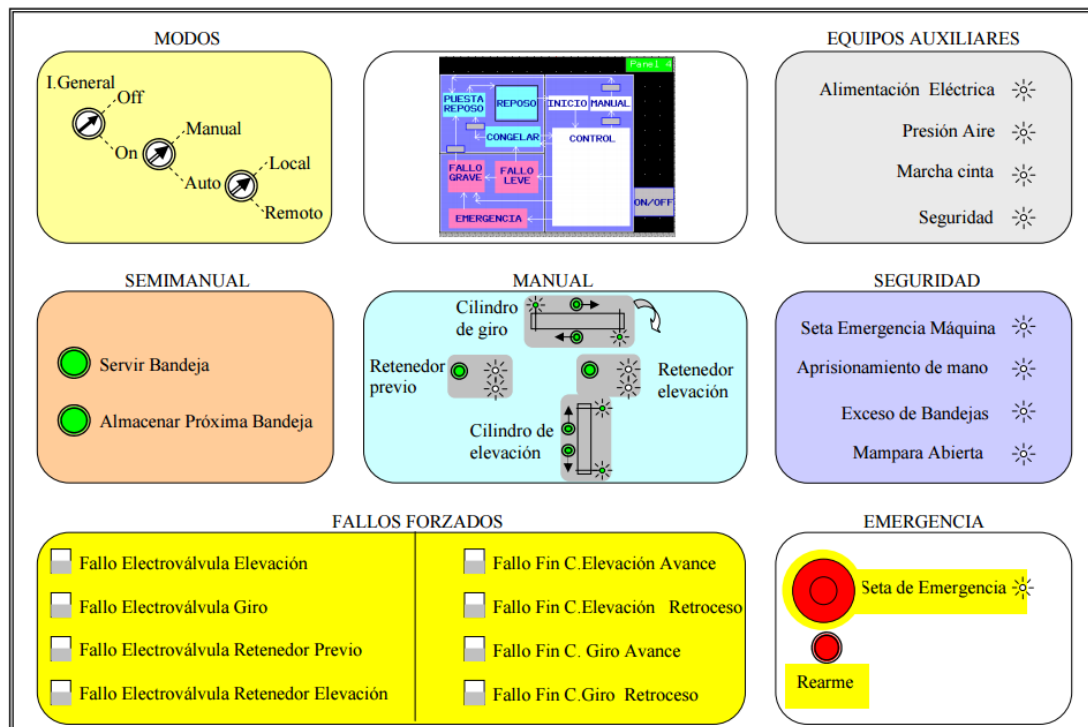


Ilustración 69: Panel de control del alimentador de piezas.

7.2.1.2.1 Modos

Consta de tres selectores que permiten seleccionar los modos de funcionamiento. El primer selector realiza la conexión de la máquina a la red, mientras que las combinaciones del segundo y tercer selector permiten elegir entre modo de funcionamiento manual, funcionamiento automático local, o funcionamiento automático remoto.

7.2.1.2.2 Pantalla táctil

Permite la supervisión y la interacción con el sistema de forma intuitiva. La programación de la interfaz de esta pantalla no entra dentro de los objetivos de este proyecto, por lo que no se especificarán más detalles sobre ella.

7.2.1.2.3 Indicadores de equipos auxiliares

Se dispone de una serie de indicadores que informan del correcto funcionamiento de los equipos auxiliares. Para el funcionamiento del alimentador se necesitará disponer de alimentación eléctrica, el movimiento activo de la cinta del tramo 4 y tener presión de aire, así como el relé de seguridad activo.

7.2.1.2.4 Pulsadores del modo semimanual

Consta de dos pulsadores, correspondientes a las órdenes de almacenar o servir bandeja respectivamente (válidos sólo en el modo de funcionamiento automático local).

7.2.1.2.5 Modo manual

En ella se encuentran pulsadores para cada uno de los actuadores del alimentador, así como indicadores luminosos que indican el estado actual de los sensores.

Los pulsadores e indicadores luminosos se encuentran conectados físicamente a los actuadores y sensores en el caso de que el selector se encuentre en modo manual, por lo que no es necesaria la programación de este modo.

7.2.1.2.6 Indicadores de seguridad

Muestra la causa de la parada de emergencia que se produzca en el alimentador de bandejas. Existen 4 indicadores, correspondientes a la seta de emergencia, el aprisionamiento de una mano en la máquina, la apertura de la mampara de seguridad para la manipulación de las bandejas, y exceso del número máximo de

bandejas del dispensador.

7.2.1.2.7 Forzado de fallos

Consta de ocho interruptores que permiten provocar fallos en los finales de carrera de los pistones de elevación y giro, así como el fallo de las electroválvulas de los actuadores de la máquina (cilindro de elevación, de giro, y retenedores previo y de elevación).

7.2.1.2.8 Emergencia

En ella se encuentra la seta de emergencia, así como el botón *Rearme* que activa el relé de seguridad para permitir la operación de la máquina.

7.2.1.3 Descripción de los módulos del PLC

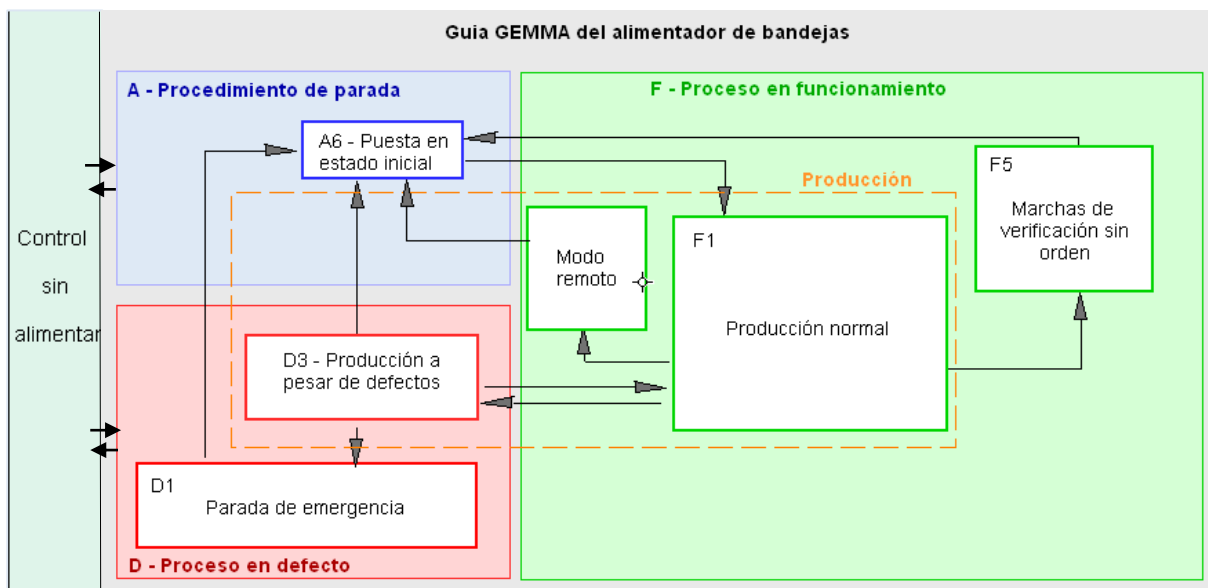
El autómatas existente en esta máquina es un autómatas Modicon M340 de Schneider Electric, configurado con los siguientes módulos:

- Módulo CPU P34 2020 2.0: Consta en el exterior de una pantalla de indicadores LEDs, un puerto USB, un puerto Ethernet, una ranura para una tarjeta de memoria y un puerto serie MODBUS. En los capítulos 4 y 5 se han descrito con mayor profundidad los aspectos físicos referidos a la comunicación.
- Tarjeta 16 Entradas digitales DDI1602: Módulo binario de 16 entradas digitales
- Tarjeta 8 Entradas digitales/8 salidas digitales DDM16022: Módulo binario de 8 entradas y 8 salidas.

El autómatas de esta máquina será considerado como el PLC secundario en la instalación. Éste se encuentra conectado mediante una conexión Ethernet al PLC que coordinará toda la instalación (autómatas principal).

7.2.2 Presentación de la guía GEMMA desarrollada

A continuación, se presenta la guía GEMMA desarrollada y la explicación de los distintos estados posibles.



7.2.2.1 A6 - Puesta en estado inicial

En este estado se llevará el alimentador al estado inicial: manteniendo el retenedor previo activado se subirá el pistón de elevación a su posición superior (con el fin de subir cualquier bandeja que se encuentre atascada impidiendo el movimiento del cilindro de giro), y tras esto se activarán las electroválvulas que mueven los pistones a la posición de reposo (exceptuando el pistón de giro).

7.2.2.2 F1- Producción normal

Corresponde al modo de funcionamiento automático local (puede ser seleccionado usando los selectores disponibles en el panel de control). Las peticiones son dadas con los pulsadores situados en el panel de control del alimentador (servir y almacenar bandejas). Estas peticiones serán acumulativas (es decir, si el pulsador de servir bandeja es pulsado tres veces entonces almacenará 3 peticiones de servir bandeja) y sólo serán realizadas en caso de que sea posible (si hay tres peticiones de servir bandejas pero sólo una bandeja, se llevará a cabo una petición, y las dos peticiones restantes quedarán pendientes para el momento en las que las pueda llevar a cabo).

7.2.2.3 F5 - Modo de verificación sin orden

Conocido también como modo manual, puede ser accedido con el selector del panel del control. Permite el manejo de los distintos actuadores a través de los pulsadores ubicados en el panel de control. En este modo los pulsadores e indicadores están directamente conectados a los actuadores, de forma que el PLC no debe actuar sobre los actuadores.

Una vez que se haya terminado de operar en este modo, el único estado posible será el *A6 Puesta en marcha*, con el fin de llevar el alimentador al estado inicial.

7.2.2.4 Modo remoto

Este modo no se encuentra contemplado dentro de la guía GEMMA, por lo que se ha incorporado como un estado adicional. Las peticiones serán escritas de forma remota en palabras internas del PLC por el autómata principal usando la conexión Ethernet.

El funcionamiento es idéntico al del alimentador en el estado de producción normal.

7.2.2.5 D3 - Producción a pesar de defectos

Consiste en la producción normal o la producción en modo remoto cuando se ha producido algún error leve. En este estado, la producción sigue en proceso, aunque no de la forma más óptima posible.

Los fallos leves contemplados son el fallo de sensores de finales de carrera de los cilindros de elevación o de giro. En el caso de solventarse el fallo leve, el alimentador regresa de nuevo al modo de funcionamiento normal A1.

7.2.2.6 D1 - Parada de emergencia

Corresponde al estado de la máquina cuando se ha producido un fallo grave o cuando el relé de seguridad se haya desactivado.

El relé de seguridad se desactiva con la pulsación de la seta de emergencia, la apertura de la mampara, el aprisionamiento de algún objeto con el cilindro o un exceso de bandejas en el alimentador. Se considera como falo grave de la máquina el fallo de los actuadores.

Con el fin de salir de este estado tras un fallo grave, se debe pulsar la seta de emergencia. Tras esto, indicar que el fallo ha sido reparado mediante la pulsación de los pulsadores de *Almacenar bandeja* y *Servir bandeja*. Una vez reparado el fallo, se podrá introducir el número actual de bandejas en el almacén de forma manual⁶. Para ello, se pulsan los dos pulsadores al mismo tiempo teniendo el selector del modo en *AUTO*, y se marcan el número de bandejas dando flancos de bajada con el selector de *modo local/remoto* tantas veces como piezas haya. Por último, se desactiva la seta de emergencia, y se pulsa el pulsador de *Rearme*.

7.2.3 Implementación usando Unity Pro XL

En este apartado se presentan la lista de variables de los módulos de entrada y salida con sus respectivas direcciones, así como las secciones llevadas a cabo para la programación.

⁶ La reparación del fallo y el ajuste del número de bandejas también puede ser realizado desde la aplicación SCADA.

7.2.3.1 Configuración del PLC

Primero se deben introducir en el programa los distintos módulos que posee el PLC que se están utilizando. Para ello, en el explorador de proyectos se selecciona el apartado *Configuración* dentro del explorador de proyectos.

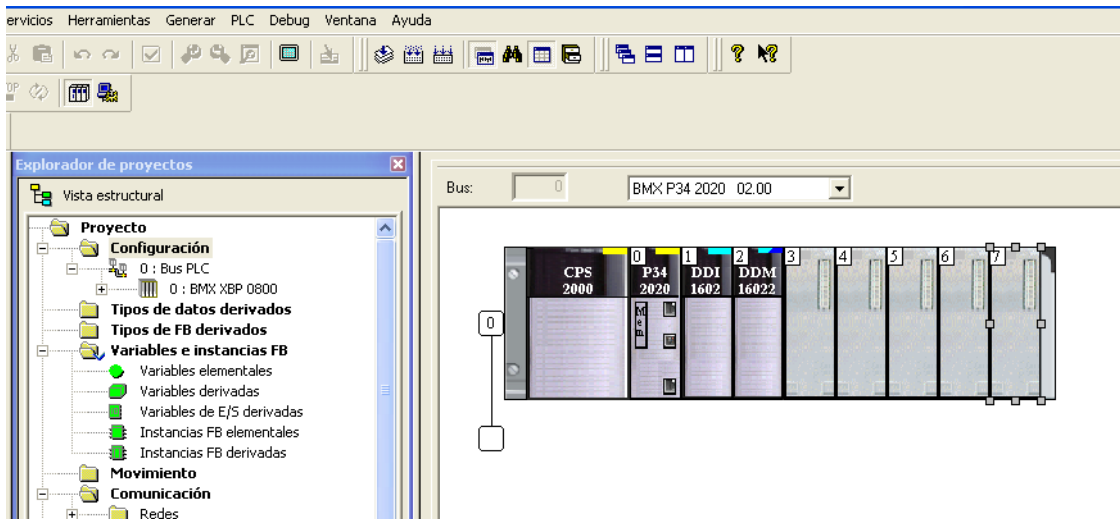


Ilustración 70: Ventana de configuración de Unity Pro XL.

En la ventana de configuración se deben seleccionar los módulos que posee el autómatas a programar. En este caso son los mismos que se indicaron en el apartado 7.2.1.3.

7.2.3.2 Direcciones de las entradas y salidas al autómatas

Variable	Entrada/salida	Tipo	Dirección
ED_S1_BAND_PREV	Entrada	EBOOL	%I0.1.0
ED_S2_BAND_ELEV	Entrada	EBOOL	%I0.1.1
ED_S3_ALTO_RET_PREV	Entrada	EBOOL	%I0.1.2
ED_S4_BAJO_RET_PREV	Entrada	EBOOL	%I0.1.3
ED_S5_ALTO_RET_ELEV	Entrada	EBOOL	%I0.1.4
ED_S6_BAJO_RET_ELEV	Entrada	EBOOL	%I0.1.5
ED_S7_SUP_CIL_ELEV	Entrada	EBOOL	%I0.1.6
ED_S8_INF_CIL_ELEV	Entrada	EBOOL	%I0.1.7
ED_S9_SUP_CIL_GIRO	Entrada	EBOOL	%I0.1.8
ED_S10_INF_CIL_GIRO	Entrada	EBOOL	%I0.1.9
ED_S11_BARRIDO	Entrada	EBOOL	%I0.1.10
ED_P1_AUTO_MANUAL	Entrada	EBOOL	%I0.1.11
ED_P2_LOCAL_REMOTO	Entrada	EBOOL	%I0.1.12

ED_P3_EMERG	Entrada	EBOOL	%I0.1.13
ED_P4_ALMAC_BAND	Entrada	EBOOL	%I0.1.14
ED_P5_SERVIR_BAND	Entrada	EBOOL	%I0.1.15
ED_RELE_SEG	Entrada	EBOOL	%I0.2.0
ED_M1_SEG_MANOS	Entrada	EBOOL	%I0.2.1
ED_M2_SEG_MAX_BAND	Entrada	EBOOL	%I0.2.2
ED_M3_SEG_MAMPARA	Entrada	EBOOL	%I0.2.3
ED_M4_SEG_SETA	Entrada	EBOOL	%I0.2.4
ED_EX2_PRESOSTATO	Entrada	EBOOL	%I0.2.5
SD_A1_BAJA_CIL_ELEV	Salida	EBOOL	%Q0.2.16
SD_A2_SUBE_CIL_ELEV	Salida	EBOOL	%Q0.2.17
SD_A3_MIN_CIL_GIRO	Salida	EBOOL	%Q0.2.18
SD_A4_MAX_CIL_GIRO	Salida	EBOOL	%Q0.2.19
SD_A5_RET_PREV	Salida	EBOOL	%Q0.2.20
SD_A6_RET_ELEV	Salida	EBOOL	%Q0.2.21

7.2.3.3 Programación

La programación propuesta está formada sólo por una única tarea maestra, la cual se compone de 11 secciones que se encargan de distintos aspectos del alimentador. A continuación, se presentará qué es lo que hace cada una de las secciones, las cuales se pueden encontrar en el Anexo B.

7.2.3.3.1 Gestor_peticiones (LD)

Esta sección actualiza las peticiones de almacenamiento y de servir bandeja durante el funcionamiento del alimentador. Distingue entre peticiones remotas (cuando se tiene seleccionado el modo automático remoto) y peticiones locales (cuando se tiene seleccionado el modo automático local).

En el caso del modo local, las peticiones aumentan con los pulsadores del panel de control mientras que se decrementan con la ejecución de la petición correspondiente.

En el caso del modo remoto sólo se llevan a cabo los decrementos, puesto que el número de bandejas a servir o almacenar será escrito por el PLC principal. Con el fin de conocer cuándo se acaba un ciclo de almacenado o extracción de bandeja, se usan dos variables bandera que serán activadas por la sección *Modo_control*.

Esta sección también se encarga de actualizar el número de bandejas actual del almacén: si se almacena una bandeja, se incrementa en uno el número de bandejas; si se extrae una bandeja, se decrementa en uno el número de bandejas. Este número es reseteado en el caso de un flanco de subida del relé de seguridad (tras la salida de una parada de emergencia en la que el relé de seguridad haya sido desactivado).

7.2.3.3.2 Guía_GEMMA (SFC)

Esta sección es la implementación de la guía GEMMA presentada al principio de este capítulo. Consiste en un SFC que coordina la activación de las distintas secciones.

7.2.3.3.3 Supervisor (LD)

Realiza la detección de los errores, y activa las transiciones en la guía GEMMA.

El supervisor distingue dos tipos de fallos: fallos leves y fallos graves.

- Se considera fallo leve cuando se produzca el fallo de alguno de los sensores de final de carrera del cilindro de elevación o de giro.
- Se considera fallo grave el fallo de alguna de las electroválvulas de los actuadores del sistema (retenedores, cilindro de giro o cilindro de elevación) o la pérdida de la presión de aire.

Con el fin de que sean detectados, cada fallo posible tiene asociado un temporizador que establece el tiempo máximo para la realización de esa tarea. En el caso de que estos temporizadores sean activados, se activa el fallo correspondiente.

El reseteo de los fallos dependerá de su tipo:

- Los fallos leves serán desactivados en el caso de que se realice de nuevo cualquier operación usando esos sensores de forma correcta.
- Los fallos graves causados por fallo de los actuadores deben ser desactivados de forma manual en el panel de control pulsando los dos pulsadores de petición a la vez mientras se encuentre la seta de emergencia activa. También es posible la desactivación de forma remota desde el sistema SCADA.

7.2.3.3.4 Modo_control (SFC)

Sección encargada de llevar a cabo las peticiones. Corresponde en la guía GEMMA a los estados F1-*Producción normal*, D1-*Producción a pesar de defectos* y al *modo remoto*.

Del paso inicial aparecen dos bifurcaciones: la primera corresponde a si hay petición de almacenamiento, mientras que la segunda si hay petición de servir bandeja.

Para almacenar bandeja, el retenedor de bandejas en zona previa es activado durante 3 segundos, con el fin de esperar a la salida de cualquier bandeja que se encuentre en su interior. Tras esto, si existe una bandeja esperando en la zona previa, se desactiva el retenedor para dejar pasar la bandeja a la zona de elevación, y se vuelve a subir con el fin de retener a aquellas que vengan tras ella. Para parar las bandejas en la zona de elevación, se activa el retenedor de bandeja en posición de elevación. Una vez es detectada por el sensor de la zona de elevación, en el caso en el que el sensor de barrido haya detectado un pallet en la bandeja se desactiva el retenedor, dejando paso a una nueva bandeja. Si el sensor de barrido no detecta pallet sobre la bandeja, se activa el cilindro de elevación hasta que llegue a su posición superior. Tras esto, se procede al giro usando el actuador para ello (deberá chequearse antes la posición del cilindro con el fin de saber cuál de los actuadores activar). Por último, se procede a bajar el cilindro de elevación y el retenedor de bandejas en posición de elevación, y se indica con una variable tipo bandera que el ciclo de almacenado ha finalizado. En el caso de que haya más peticiones de almacenamiento, el retenedor de bandeja previa se deja elevado; si por el contrario ya no quedan más peticiones, se procede al descenso del retenedor.

Para servir bandeja, se activa el retenedor de bandejas en zona previa durante 5 segundos para esperar a la salida de cualquier bandeja que se encuentre en su interior. Tras esto, se procede a activar el cilindro de elevación hasta que llegue a la posición superior, y se procede al giro usando el actuador para ello (se debe chequear antes la posición del cilindro con el fin de saber cuál de los actuadores activar). Una vez se haya girado, se procede a bajar el cilindro de elevación y tras esto, desactivar el retenedor de bandeja en zona de elevación. Por último, se notifica que el ciclo de servir bandeja ha acabado con una variable tipo bandera, y se desactiva el retenedor de bandeja previa.

7.2.3.3.5 Puesta_en_reposo (SFC)

Realiza la puesta en el estado inicial, por lo que corresponde al estado A6 de la guía GEMMA.

Esta secuencia es activada por una variable tipo bandera. En ella se llevan todos los pistones a sus posiciones

iniciales, y tras ello, se eleva el cilindro de elevación a su posición superior durante unos segundos con el fin de corregir cualquier defecto de posición de las bandejas.

Además de la secuencia comentada, también lleva a cabo el reseteo de las peticiones y de la sección de *Modo_control* explicada anteriormente.

7.2.3.3.6 Secuencia_emergencia (SFC)

Esta sección lleva a cabo una secuencia de acciones necesaria antes de llevar la instalación a la parada de emergencia, con el fin de que no se atasquen las bandejas en el caso de fallo grave en el cilindro de giro.

La secuencia consiste en mantener el retenedor de bandeja previa subido mientras que se hace descender el cilindro de elevación.

7.2.3.3.7 Reseteos_secciones (LD)

Si un autómatas se encontraba en ejecución mientras se ha desconectado, cuando se vuelva a encender seguirá la ejecución por el mismo punto donde quedó. No obstante, esto puede provocar situaciones peligrosas si las condiciones de la máquina fueron alteradas mientras estuvo apagada.

Por ello, se necesita que cuando se vuelva a encender comience de nuevo en el estado A6 de la guía GEMMA.

Con ese objetivo, en esta sección se llevará a cabo el reseteo de todas las secciones en el primer ciclo del autómatas tras su encendido. Para este fin se usa como condición del reseteo el bit del sistema %S21, que se activará sólo durante el primer ciclo del autómatas.

7.2.3.3.8 SCADA_Asignacion_variables(LD)

En esta sección se hace la asignación de las variables direccionadas en los módulos de entrada y salida a bits internos del autómatas, con el fin de poder ser tratadas en el SCADA. Esto es consecuencia del uso de la comunicación Unity Speedlink, que sólo permite la manipulación de bits y palabras direccionadas en la memoria.

7.2.3.3.9 Generacion_pulsos_Visual (LD)

Consta de un bloque Sample_TM que genera pulsos cada 500 ms. Este generador de pulsos es usado para la animación del SCADA.

7.2.3.3.10 Visualizacion_SCADA (ST)

En esta sección se crean las variables usadas para la representación del movimiento en el SCADA. Para ello, se usan los pulsos del generador de pulsos para contar el tiempo que pasa durante la realización de la elevación y descenso del pistón, o el movimiento de la bandeja desde el retenedor a la zona de elevación.

Estas variables son de tipo INT, y son usadas para el movimiento de la bandeja en la pantalla correspondiente del SCADA.

7.2.3.3.11 Reseteos_secciones (LD)

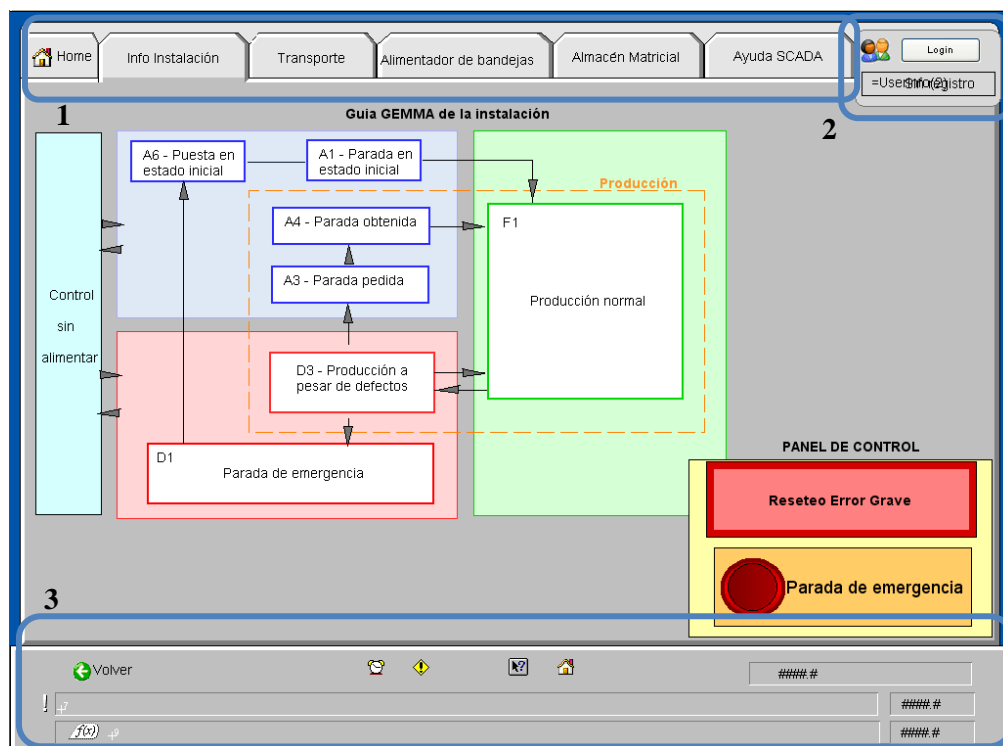
En esta sección se implementa la inicialización de las secciones SFC con el encendido del autómatas usando el bit del sistema '%S21' (este bit se encuentra a nivel alto sólo en el primer ciclo del autómatas).

También será usado para el reseteo de la sección de Modo_Control (SFC) desde el bloque de la guía GEMMA.

8 APLICACIÓN PRÁCTICA: SISTEMA DE SUPERVISIÓN SCADA

Se ha realizado una aplicación SCADA con el fin de realizar la supervisión y control de la instalación. La aplicación se ha realizado con Vijeo Citect 7.0, un software perteneciente a Schneider Electric. En este apartado se presenta el aspecto general de cada una de las ventanas de la interfaz gráfica. Tras esto, se enumeran las alarmas que posee la aplicación y, por último, se hablará del registro de la actividad en un documento de texto. Para el diseño de la interfaz gráfica se ha usado como referencia [16].

8.1 Apariencia de la ventana



Se compone de 3 elementos principales:

- Selección de ventanas (elemento 1): Permite acceder a cada una de las ventanas que componen el SCADA. Cada ventana está dedicada a una parte de la instalación.
- Selección de usuario: Permite la identificación del usuario que se encuentre usando el SCADA, de forma que las acciones que realice queden grabadas en un documento de texto.
- Barra de información general: Muestra información general de la aplicación: hora, fecha y nombre de la ventana actual o ultima alarma activada. Permite también el acceso a las

ventanas de alarmas pulsando los iconos situados a la izquierda (corresponden con las alarmas de señal y de hardware respectivamente), activar el modo de ayuda, volver a la ventana anterior o regresar a la ventana Home de la aplicación.

8.2 Pantallas principales

La aplicación principal se divide en seis pantallas, que son explicadas en los siguientes apartados. Para mayor claridad de las pantallas, parte de la información se muestra o se oculta en función del estado de la instalación.

Los colores usados siguen la convención del *Real Decreto 485/1997, de 14 de abril, sobre disposiciones mínimas en materia de señalización de seguridad y salud en el trabajo* [17].

8.2.1 Home

Presenta un resumen del estado actual de toda la instalación.

1. Presenta el estado actual de cada uno de los elementos: estado del alimentador, sistema robotizado de almacenamiento y sistema de transporte. El color del recuadro se muestra verde, amarillo o rojo en función del estado de la instalación (verde para estado sin fallos, amarillo para estado con fallos leves, o rojo para estado con fallos graves) como se aprecia en Ilustración 71.

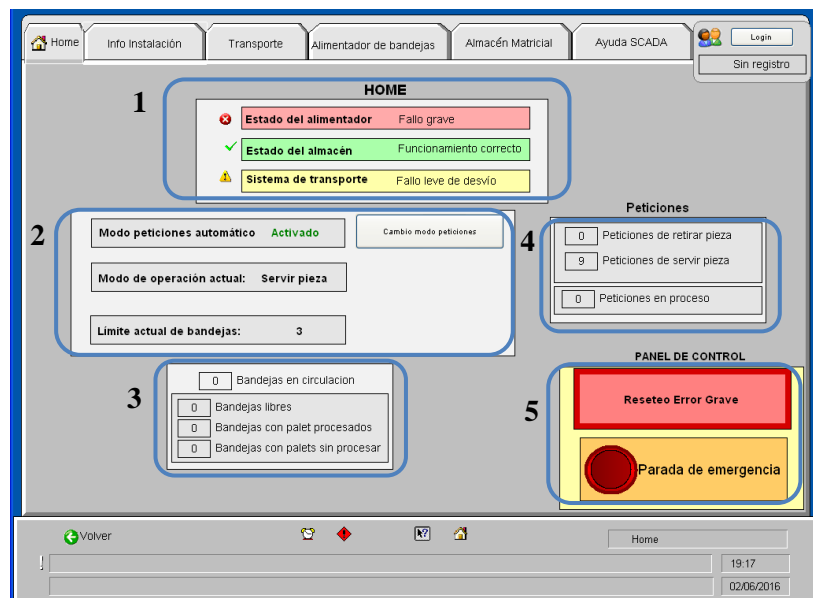


Ilustración 71: Pantalla Home con distintos errores en los equipos.

2. Muestra la configuración actual de la operación (si se encuentra en modo de selección de peticiones automáticas, si la operación es servir o retirar pieza sobre el pallet, y el límite actual de bandejas). Estas opciones son configurables con los botones situados en la derecha.
3. Muestra el número de bandejas actual en la instalación.
4. Permite la selección del número de peticiones sólo en el caso de que se encuentre en modo de peticiones manuales.
5. Cuadro de mandos de la instalación. Permite la activación de los distintos modos y solicitar parada durante el modo de producción normal. En caso de emergencia, muestra un botón de reseteo de error grave, así como un icono de un pulsador de emergencia.

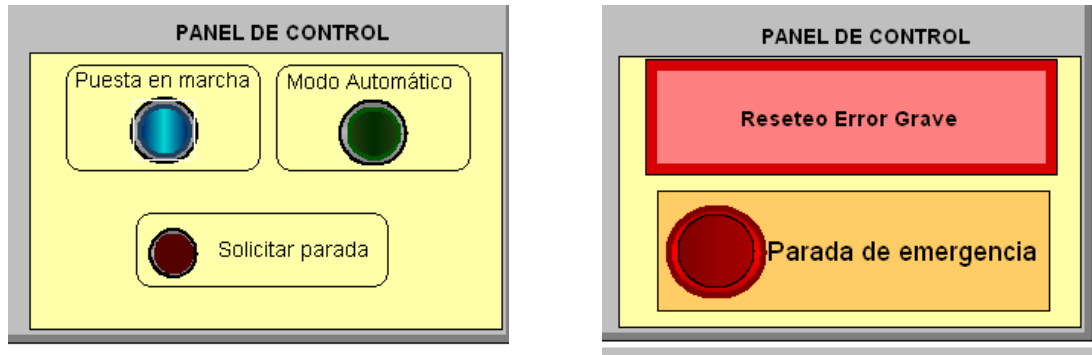


Ilustración 72: Cuadro de mandos en espera de puesta en marcha y en espera de reseteo de error grave.

8.2.2 Info instalación

Muestra la guía GEMMA de la instalación, indicando en el estado en el que se encuentra. En la esquina inferior derecha se muestra el mismo cuadro de mandos de la instalación que en la ventana de *Home*.

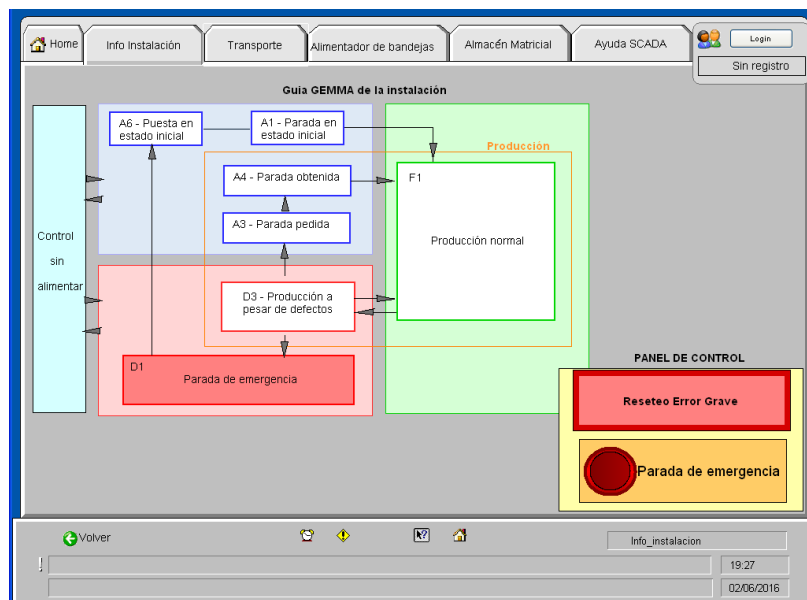


Ilustración 73: Pantalla *Info Instalación* mostrando una parada de emergencia.

8.2.3 Transporte

Muestra información relevante sobre el sistema de transporte:

- Estado de los retenedores: Se representan los retenedores elevados en color rojo, mientras que se representan en su posición inferior en color verde.
- Estado de los tramos: Se muestra una estimación de las posiciones de las bandejas en los distintos tramos, así como el estado de las bandejas (si lleva un pallet, y si éste está cargado). En el caso de fallo grave en alguno de los tramos, se muestra en el margen derecho un cuadro indicando en qué tramos se encuentra el fallo grave. En la parte superior de este cuadro, también se muestra un pulsador que permite el reseteo de este error grave.
- Estado actual de cada puesto de trabajo.
- Estado de los desvíos: En caso de fallo leve en los desvíos, se muestra una advertencia visual en el desvío correspondiente. También se representa el estado de los elevadores de los desvíos mediante su color (el color rojo implica el desvío en su posición de reposo, mientras que el verde indica el elevador elevado).

- Estado de los motores de cada tramo: En el caso de que el motor del tramo se encuentre activado, el color de la cinta es verde, mientras que si el color de la cinta es gris, el motor de ese tramo se encuentra desactivado.

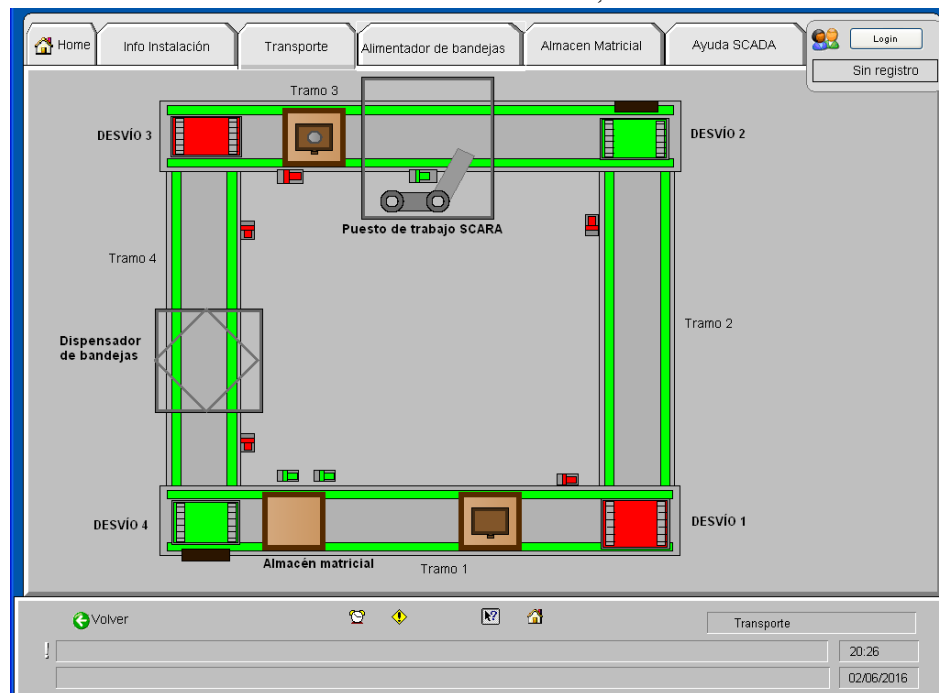


Ilustración 74: Ventana de transporte mostrando estimaciones de las posiciones de las bandejas.

8.2.4 Almacén matricial

Muestra la información del sistema de almacenaje robotizado. Se distinguen tres zonas en la interfaz:

1. Visualización del estado de ocupación del almacén: Se encuentra dividido en pallets defectuosos, pallets sin pieza y pallets con pieza. Muestra de forma visual el estado de ocupación de los estantes en tiempo real.
2. Estado del almacén: Indica en el estado en el que se encuentra el sistema robotizado de almacenaje (*ocupado, esperando petición o tarea finalizada en cinta*). Se encuentra en estado *ocupado* cuando está llevando a cabo cualquier operación (realización de una operación de almacenamiento o extracción de pallet, envío del vector de estado de ocupación, etcétera). En este caso, indica la tarea que está realizando. Cuando se encuentre el piloto *Cinta libre para mover* iluminado, la operación de manipulación del sistema de almacenaje en el recurso compartido (en este caso el tramo 1) ha finalizado.
3. Fallos graves: Indica la causa del fallo detectado. Permite resetear ambos tipos de error mediante los dos pulsadores situados en la zona derecha del cuadro. En caso de que no haya ningún fallo, los cuadros que contienen los distintos tipos de fallos no son mostrados en la ventana.

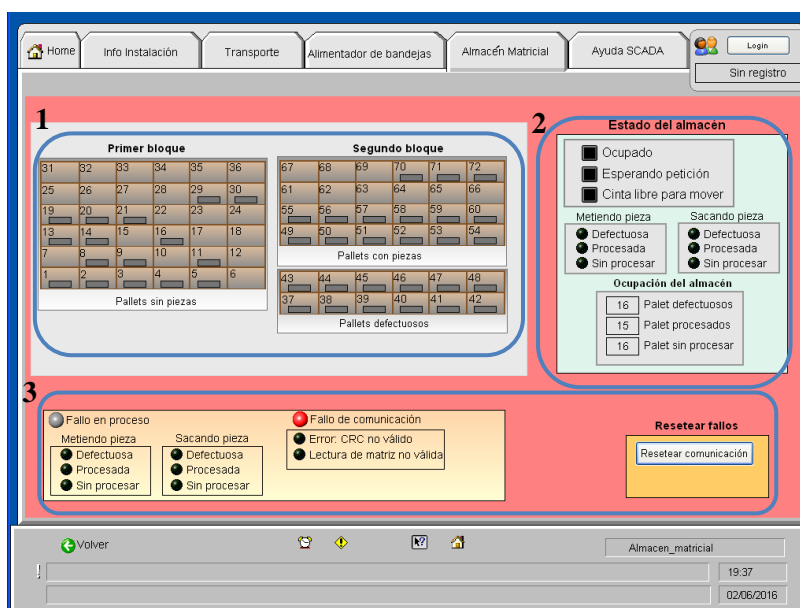


Ilustración 75: Pantalla *Almacén Matricial* mostrando un error de comunicación.

8.2.5 Alimentador de bandejas

Muestra el estado del alimentador de bandejas. En la pantalla se presentan tres zonas diferenciadas (se encuentran indicadas en la Ilustración 77).

1. Animación del estado actual del alimentador. Incluye el movimiento de los pistones y de las bandejas.
2. Información del estado actual: Se encuentra situado en el margen derecho de la pantalla. Indica el número de peticiones remotas y locales que tiene actualmente el alimentador de bandejas. También permite modificar el número de bandejas disponibles en el alimentador.
3. Fallos: Indica los fallos leves y graves producidos durante el funcionamiento del alimentador de bandejas. Son ocultados en caso de que no exista fallo de ningún tipo como se muestra en la Ilustración 76). En el caso de la existencia de un fallo grave en la máquina, se puede reparar usando el pulsador correspondiente situado sobre los errores graves.

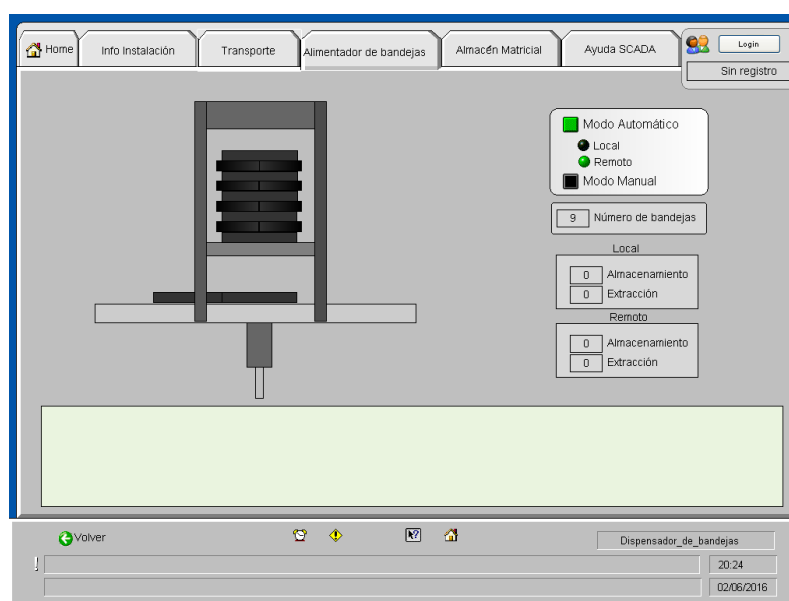


Ilustración 76: Ventana del alimentador de bandeja sin presencia de fallos.

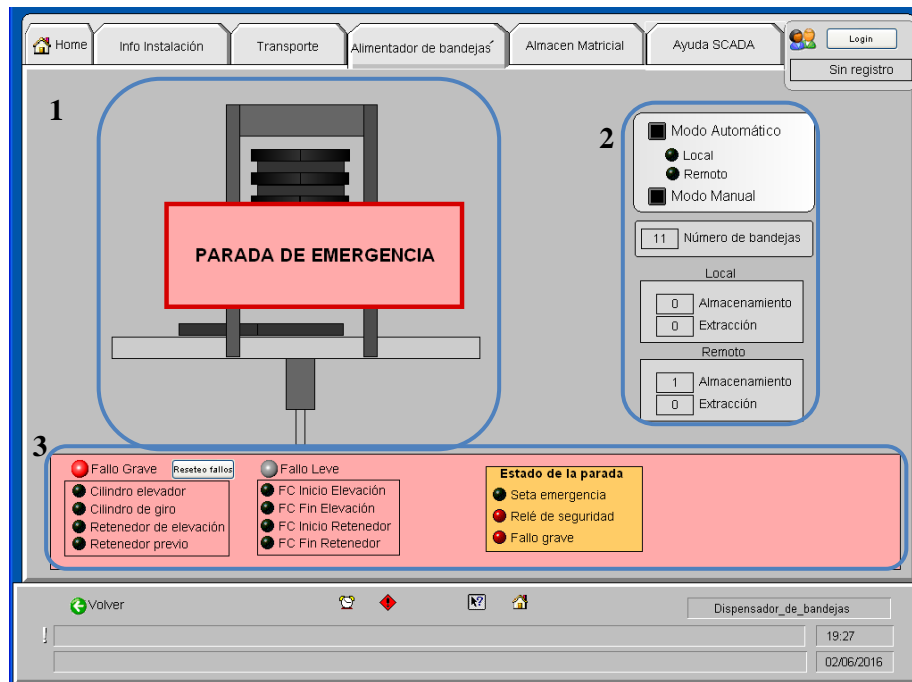


Ilustración 77: Alimentador de bandejas en caso de fallo grave.

8.2.6 Ayuda

En esta pantalla se describen brevemente los elementos de la pantalla básica de la aplicación SCADA.

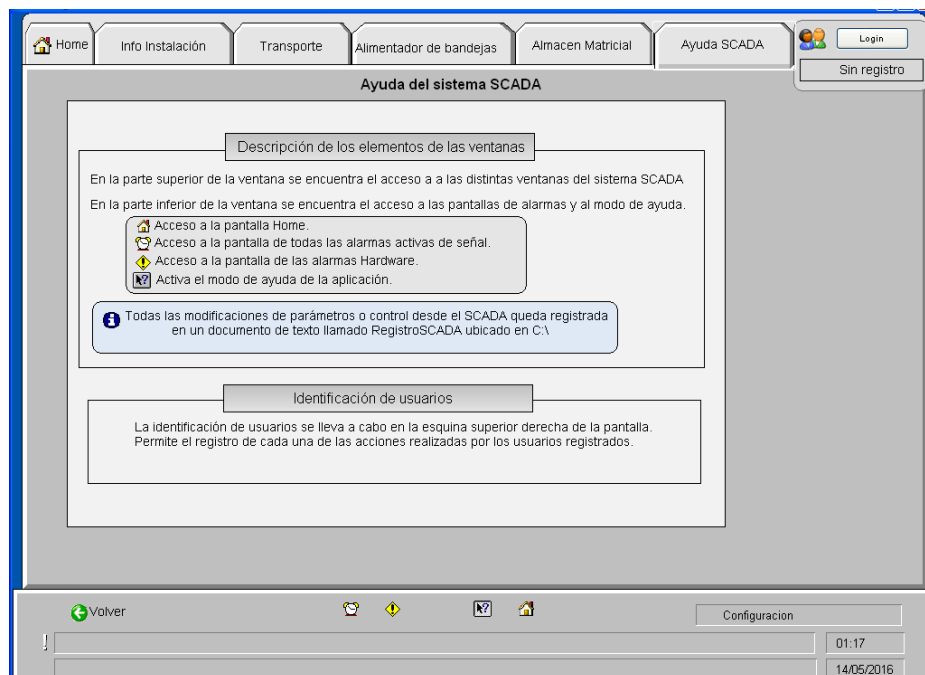


Ilustración 78: Ventana de ayuda de la aplicación SCADA.

8.3 Alarmas configuradas

En la aplicación se han configurado alarmas que alertan de situaciones anómalas en la instalación. Se distinguen dos tipos de alarmas en el sistema: hardware (por ejemplo, aquellas producidas por errores de conexión entre equipos) o digitales (producidas por la activación de una variable digital). En este caso, las alarmas digitales coinciden con los fallos graves y leves de cada uno de los equipos de la instalación.

En la Tabla 29 se encuentra un listado de las alarmas digitales implementadas, así como de sus descripciones.

Tabla 29: Alarmas con las descripciones implementadas en el SCADA.

Tag de la alarma	Nombre de la alarma	Descripción
PAR_EMERG	P. Emerg. Instalacion	Parada de emergencia en instalación.
Err_Almacen_Recep	Err. Recep. Vector	Error en recepción de vector almacén
Err_alm_defecto	Err_Alm_defecto	No hay sitio para pallet defectuoso
Err_alm_sin_tratar	Err_Alm_sin_tratar	No hay sitio para pallet sin pieza
Err_alm_tratada	Err_Alm_tratada	No hay sitio para pallet con pieza
Err_ext_tratada	Err_ext_tratada	No hay pallet con pieza
Err_ext_defecto	Err_ext_defecto	No hay pallet defectuoso
Err_ext_sn_tratar	Err_ext_sin_tratar	No hay pallet sin pieza
fallo_leve_T1	F.leve T1	Fallo leve en tramo 1
fallo_leve_T2	F.leve T2	Fallo leve en tramo 2
fallo_leve_T3	F.leve T3	Fallo leve en tramo 3
fallo_leve_T4	F.leve T4	Fallo leve en tramo 4
F_grave_T11	Fallo G. Tramo11	Desaparición bandeja en T11
F_grave_T12	Fallo G. Tramo12	Desaparición bandeja en T12
F_grave_T13	Fallo G. Tramo13	Desaparición bandeja en T13
F_grave_T14	Fallo G. Tramo14	Desaparición bandeja en T14
F_grave_T22	Fallo G. Tramo22	Desaparición bandeja en T22
F_grave_T21	Fallo G. Tramo21	Desaparición bandeja en T21
F_grave_T23	Fallo G. Tramo23	Desaparición bandeja en T23
F_grave_T31	Fallo G. Tramo31	Desaparición bandeja en T31
F_grave_T32	Fallo G. Tramo32	Desaparición bandeja en T32
F_grave_T33	Fallo G. Tramo33	Desaparición bandeja en T33
F_grave_T41	Fallo G. Tramo41	Desaparición bandeja en T41
F_grave_T42	Fallo G. Tramo42	Desaparición bandeja en T42
F_grave_T43	Fallo G. Tramo43	Desaparición bandeja en T43
F_grave_T44	Fallo G. Tramo44	Desaparición bandeja en T44

Au2_f_G_Cil_elev	Au2. F.G.Cil_elev	Fallo en cilindro de elevación
Au2_f_G_Cil_giro	Au2. F.G.Cil_giro	Fallo en cilindro de giro
Au2_f_G_Ret_Elev	Au2. F.G.Ret_Elev	Fallo en retenedor de elevación
Au2_f_G_Ret_Prev	Au2. F.G.Ret_Previo	Fallo en retenedor previo
Au2_f_L_FC2_elev	Au2. F.L. FC2 Elev	Fallo en FC2 de cil. de elev.
Au2_f_L_FC1_elev	Au2. F.L. FC1 Elev	Fallo en FC1 de cil. de elev.
Au2_f_L_FC2_Giro	Au2. F.L. FC2 Giro	Fallo en FC2 de cil. de giro.
Au2_f_L_FC1_Giro	Au2. F.L. FC1 Giro	Fallo en FC1 de cil. de giro

Para acceder a las pantallas de visualización de las alarmas, se pulsa en los iconos situados a la izquierda en la barra inferior (señalados en Ilustración 79).

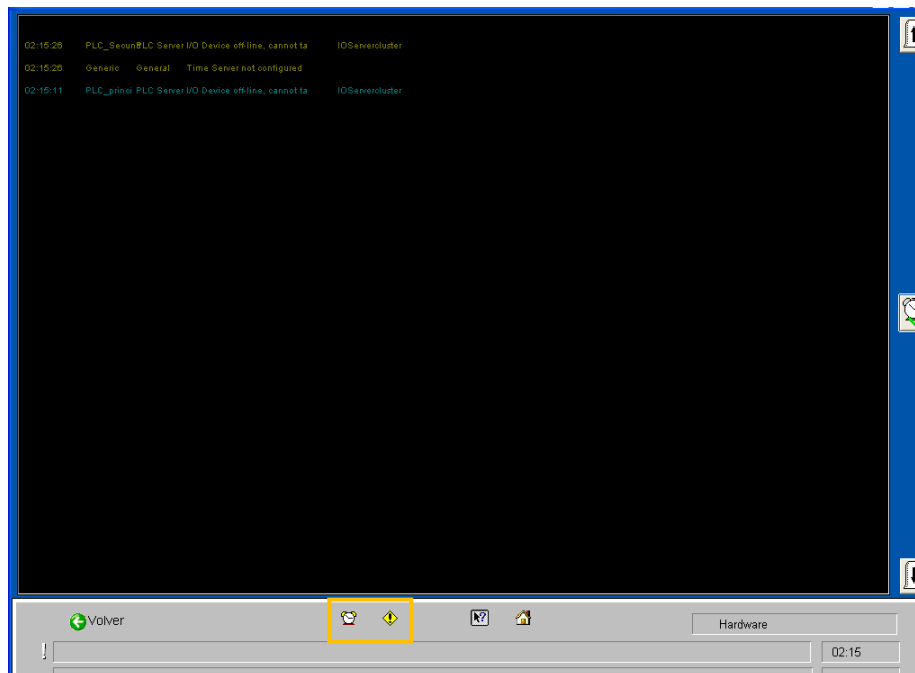


Ilustración 79: Pantalla de visualización de alarmas.

8.4 Registro de actividad en archivo de texto

La aplicación genera durante su uso un archivo de texto básico donde se indican las distintas operaciones realizadas durante la operación de la instalación desde el SCADA, así como la hora o el usuario que las ha realizado. También incorpora un registro de los distintos fallos aparecidos.

En la siguiente imagen (Ilustración 80) se puede ver un ejemplo del texto generado. Este archivo es generado en el directorio C:\ con el nombre de RegistroSCADA.

	01:11:22	Saturday, 14 de May	Home	Activación de puesta en marcha
	01:11:24	Saturday, 14 de May	Home	Activación del Estado F1
	01:11:25	Saturday, 14 de May	Home	Solicitud de parada
	01:11:26	Saturday, 14 de May	Home	Activación de puesta en marcha
	01:11:27	Saturday, 14 de May	Home	Activación del Estado F1
	01:11:27	Saturday, 14 de May	Home	Cambio modo de operación
	01:11:28	Saturday, 14 de May	Home	Cambio modo de peticiones
	01:11:28	Saturday, 14 de May	Home	Cambio modo de operación
	01:11:29	Saturday, 14 de May	Home	Cambio modo de operación
	01:11:29	Saturday, 14 de May	Home	Activación de puesta en marcha
	01:17:30	Saturday, 14 de May	Home	Solicitud de parada
	01:17:31	Saturday, 14 de May	Home	Activación del Estado F1
	01:17:32	Saturday, 14 de May	Home	Cambio modo de peticiones
	01:17:34	Saturday, 14 de May	Home	Cambio modo de operación
	01:17:37	Saturday, 14 de May	Home	Cambio modo de operación
	01:17:38	Saturday, 14 de May	Home	Cambio modo de peticiones
Profesor	01:18:36	Saturday, 14 de May	Home	Cambio modo de peticiones

Ilustración 80: Registro de la actividad en un archivo de texto.

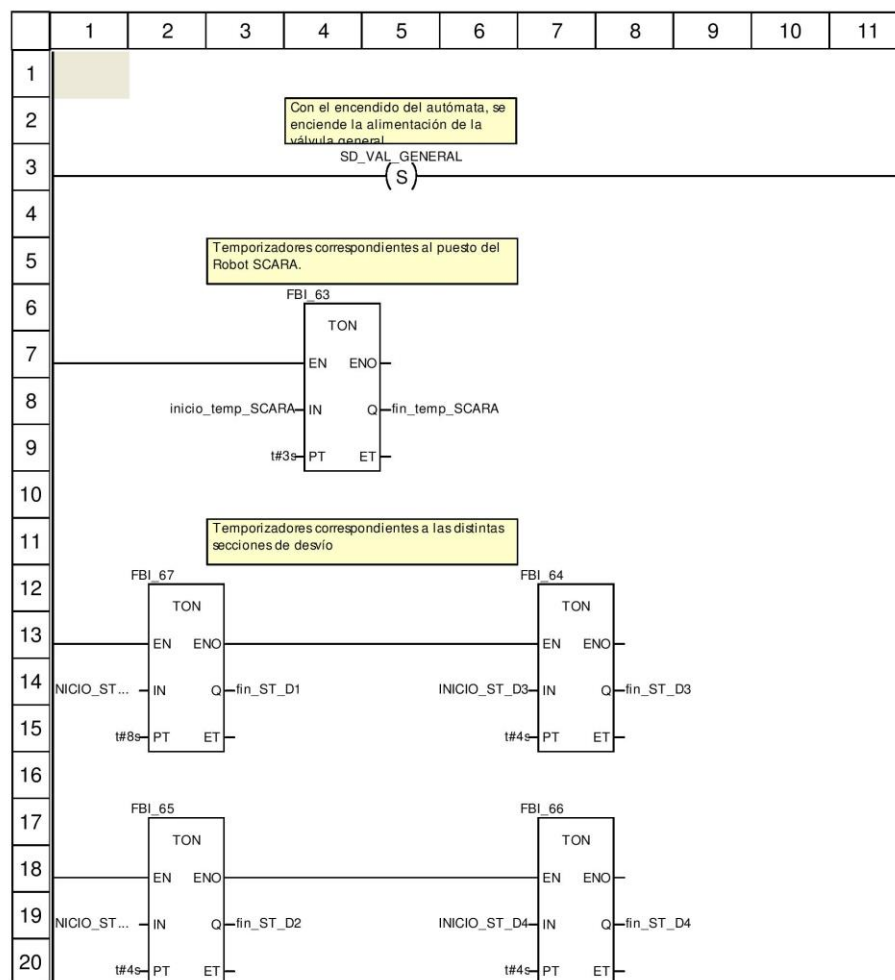
8.5 Identificación de usuarios

Como se ha comentado en la sección anterior, este sistema SCADA posee también la posibilidad de identificación del operador mediante usuarios. En esta aplicación de ejemplo se han creado dos usuarios, ambos con los mismos privilegios en el sistema. Los usuarios identificados son Alumno (con contraseña Alumno), y Profesor (con contraseña Profesor).

Esta identificación permite establecer un registro de todas las operaciones realizadas por cada uno de los operadores de la instalación.

ANEXO A: PROGRAMACIÓN DEL AUTÓMATA PRINCIPAL

valvula_general_y_Temp : [MAST]



Etiquetas truncadas:

Etiqueta	Posición(es)
INICIO_ST_D1	(1, 14)
INICIO_ST_D2	(1, 19)

Chart : [MAST - DESVIO_1]

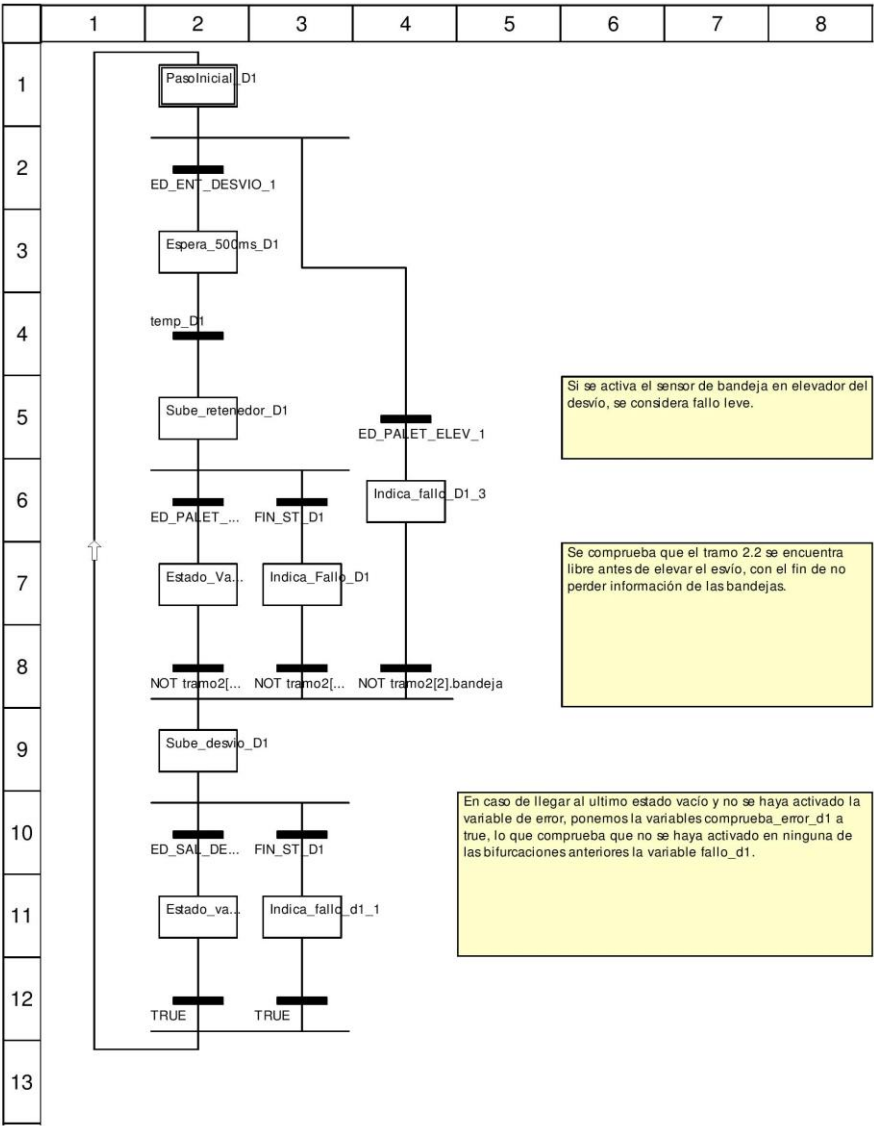


Chart : [MAST - DESVIO_2]

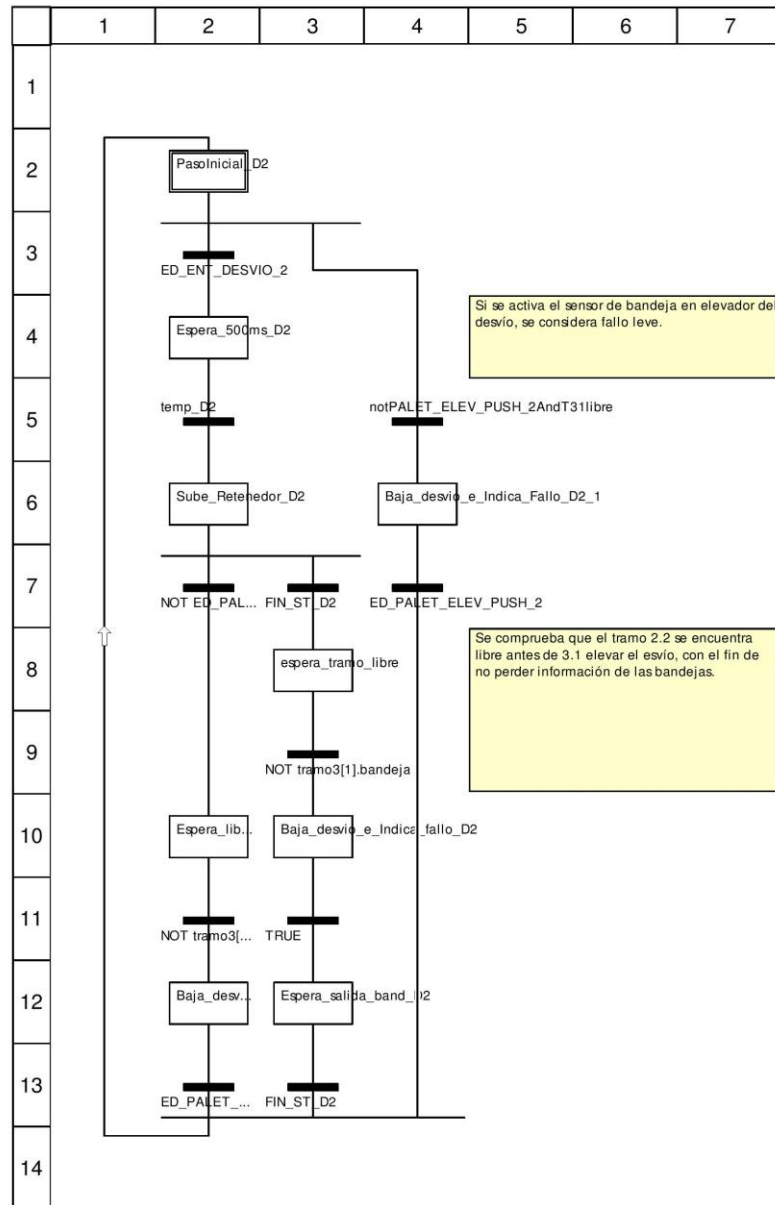


Chart : [MAST - DESVIO_3]

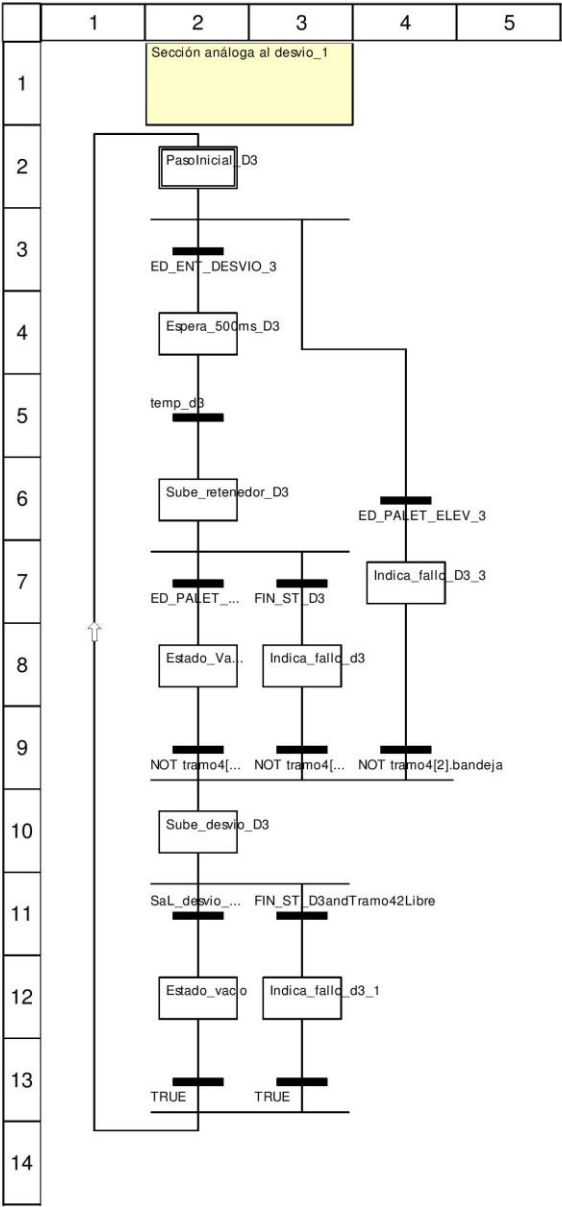


Chart : [MAST - DESVIO_4]

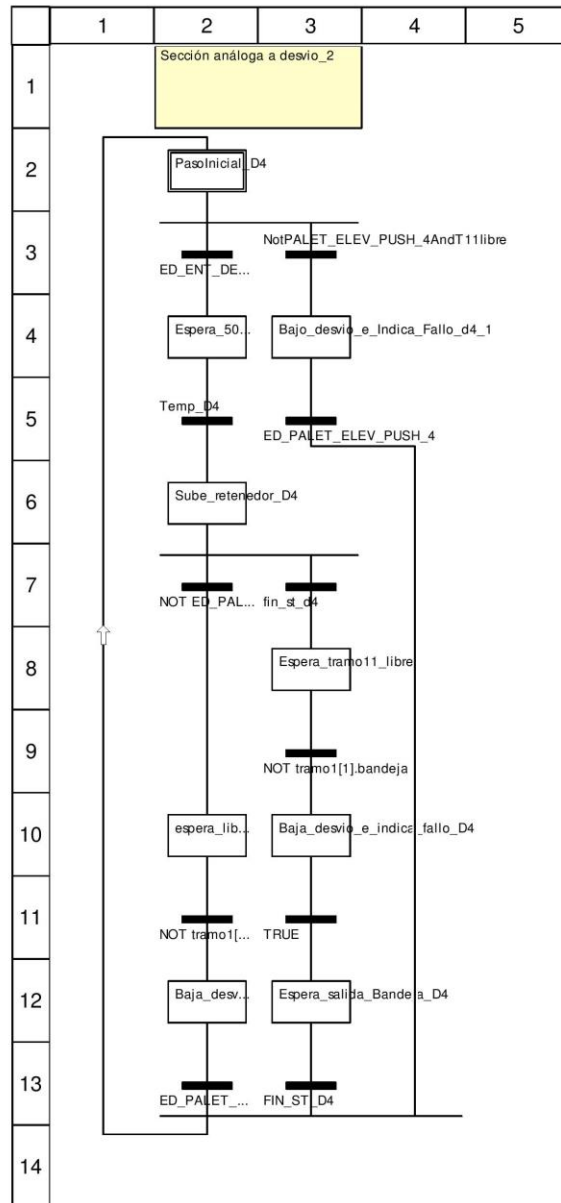
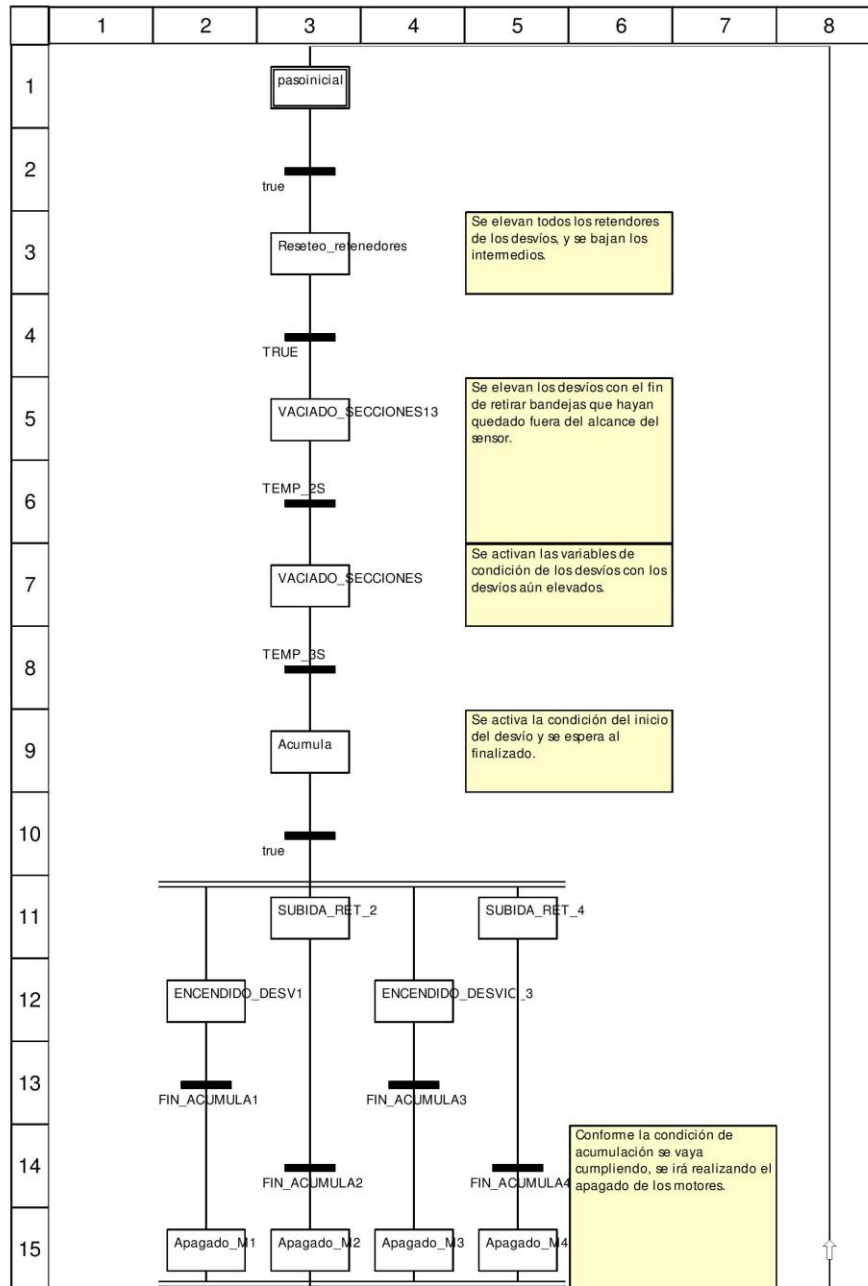
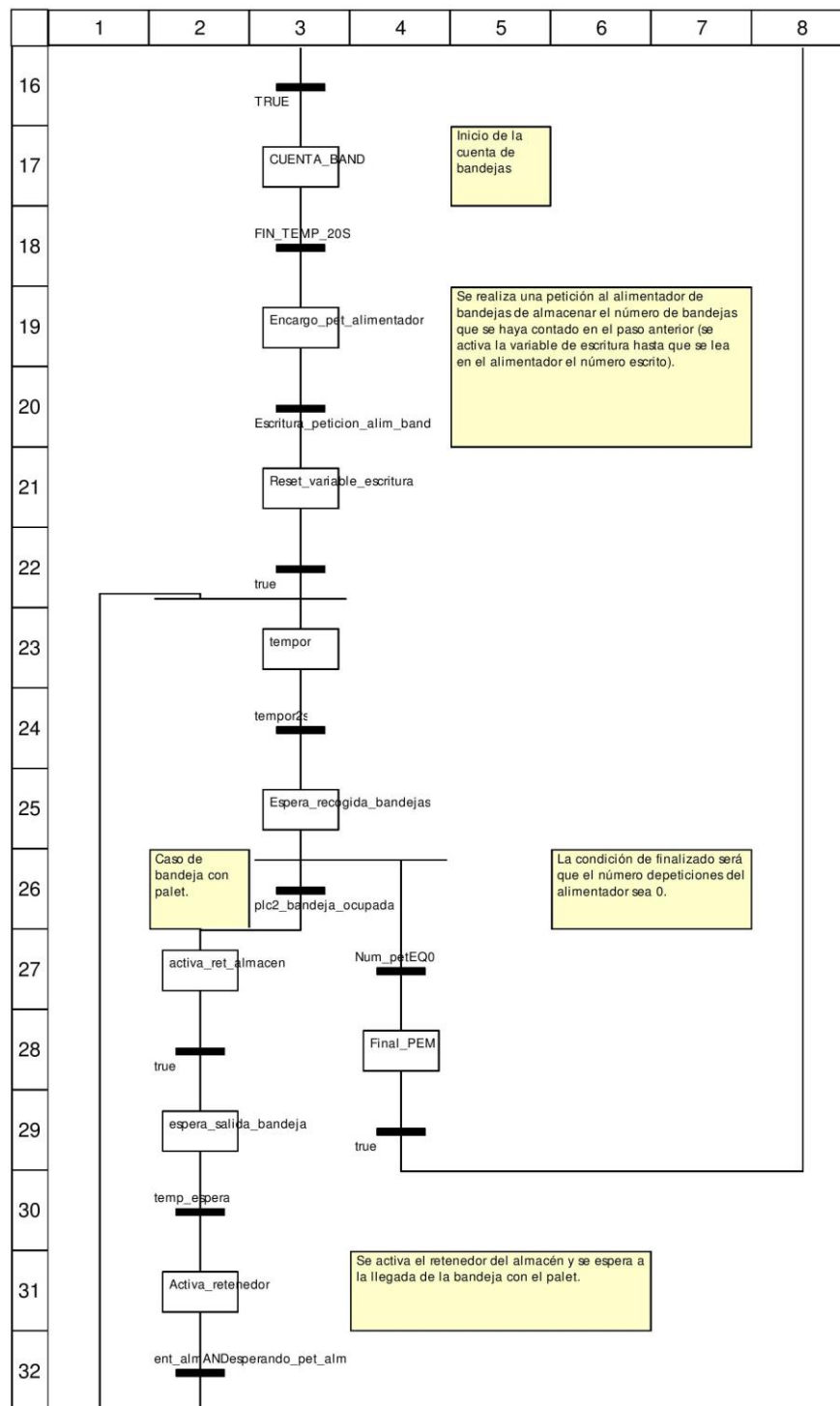
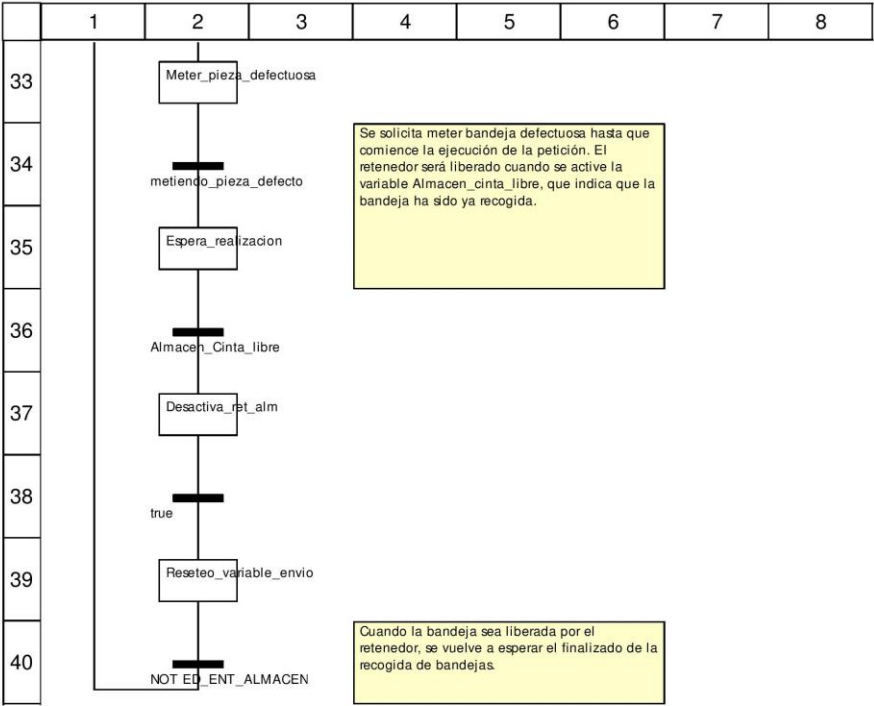


Chart : [MAST - puesta_marcha_SFC]







CONTADOR_VEH : [MAST]

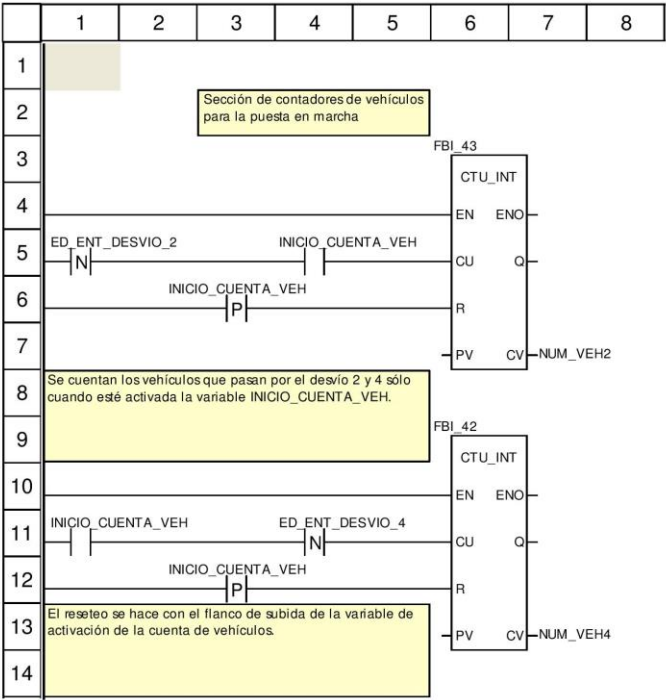
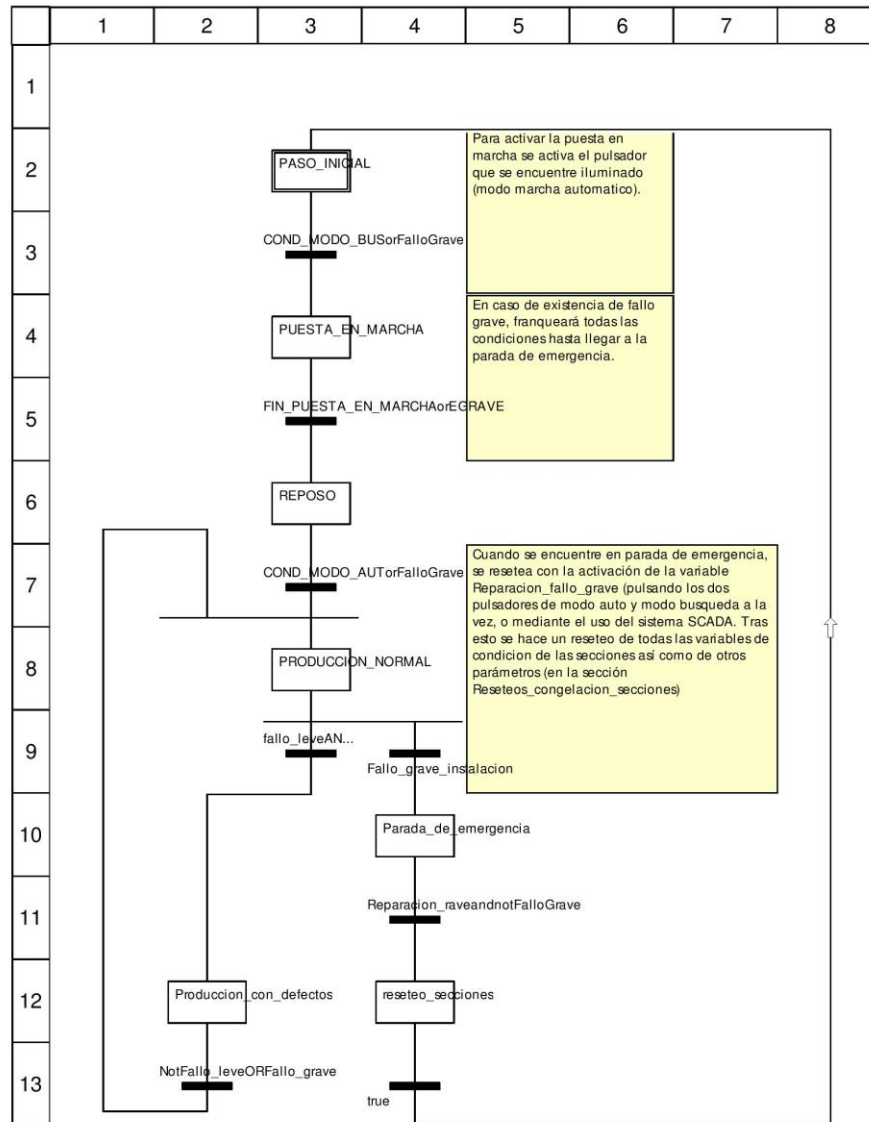
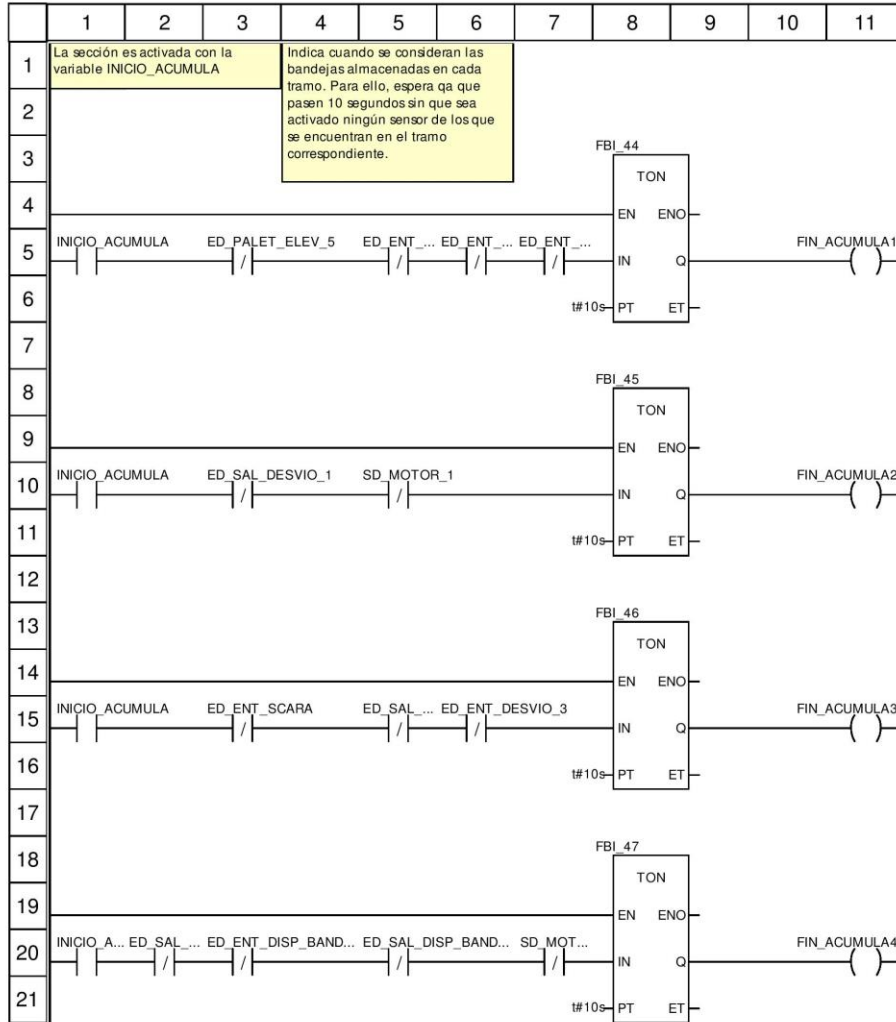


Chart : [MAST - GUIA_GEMMA]



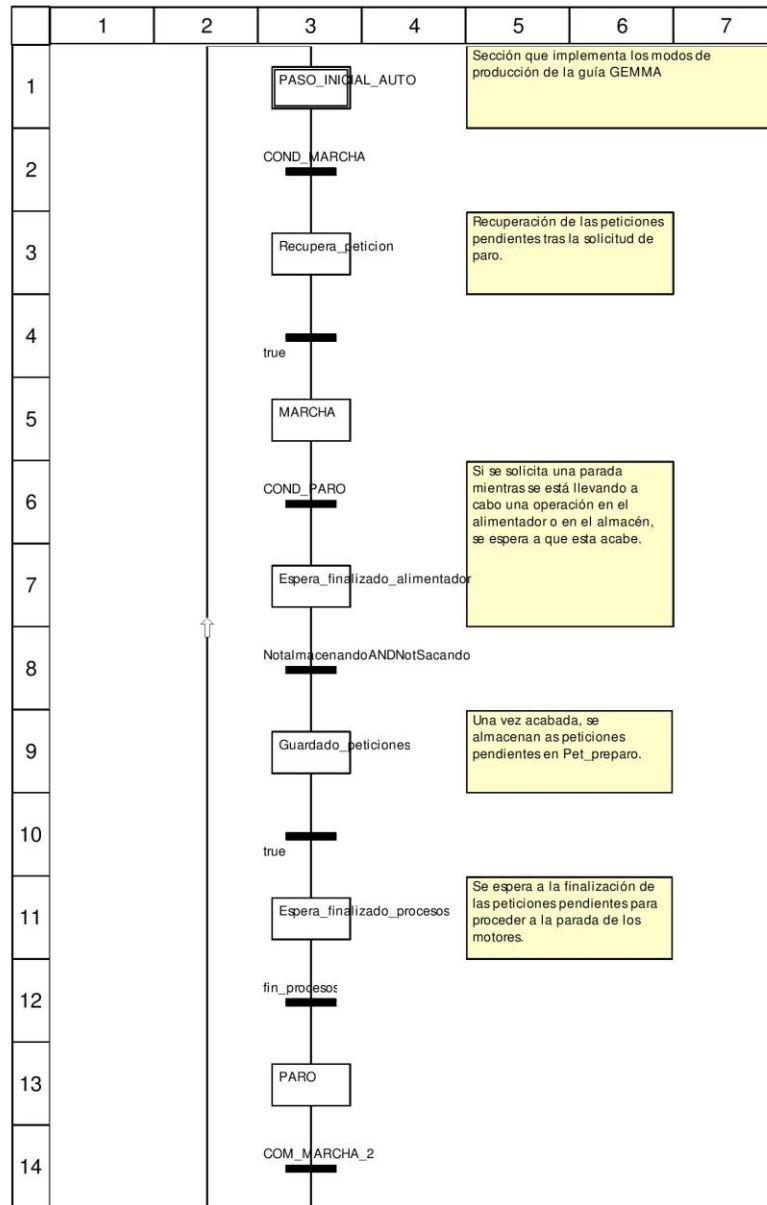
ACUMULACION : [MAST]



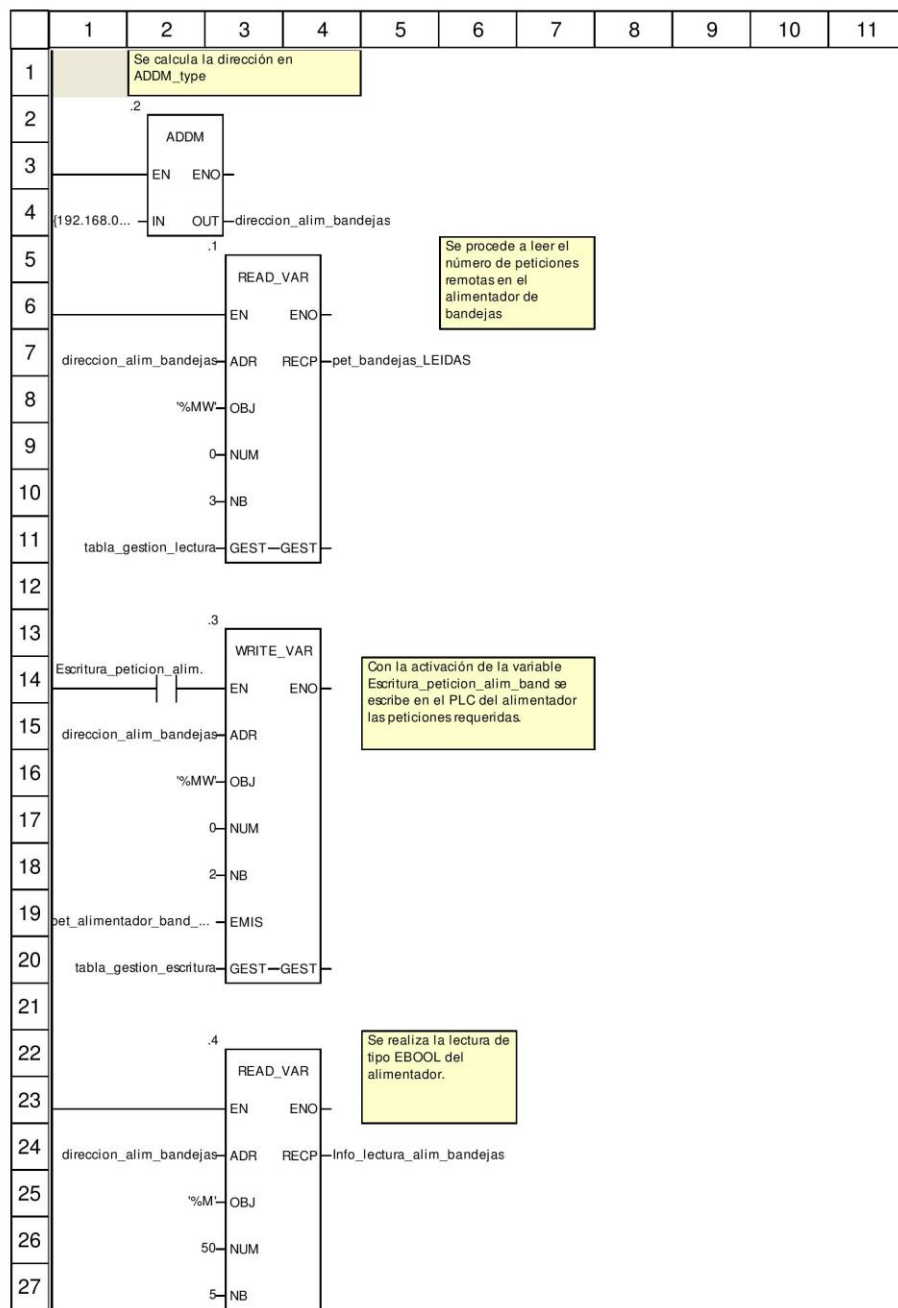
Etiquetas truncadas:

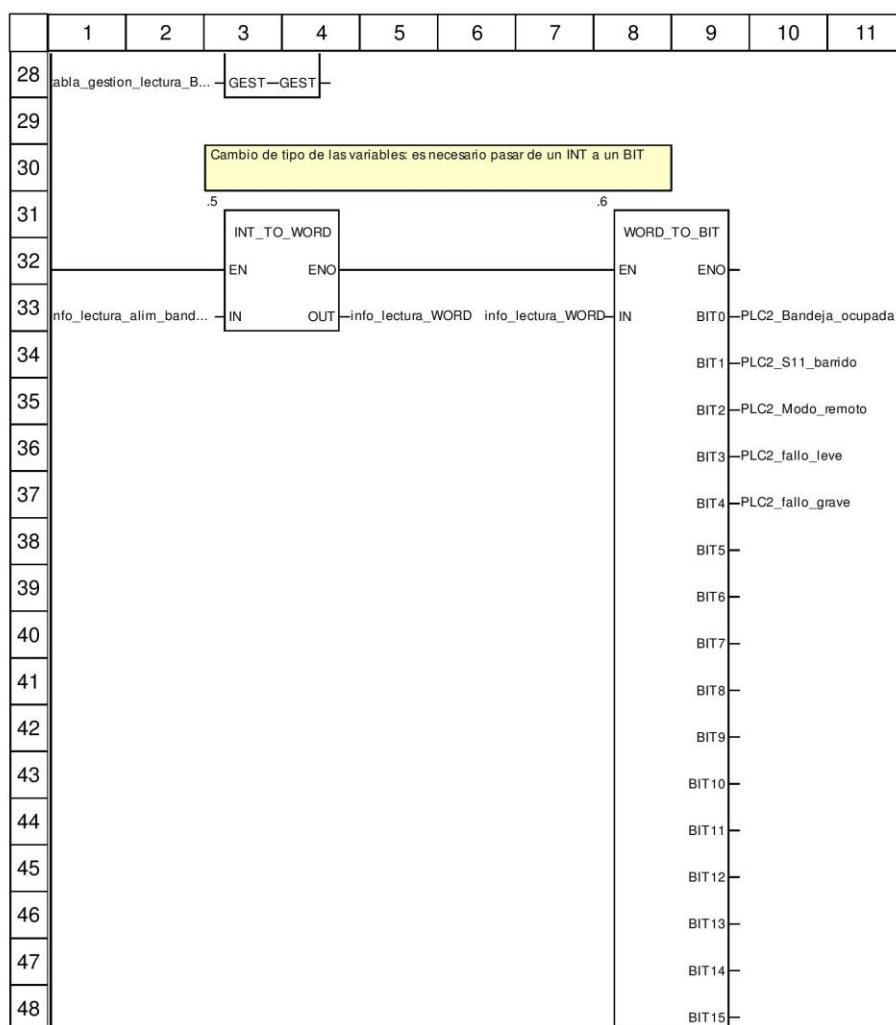
Etiqueta	Posición(es)
ED_ENT_ALMACEN	(5, 5)
ED_ENT_DESVIO_1	(7, 5)
ED_ENT_DESVIO_5	(6, 5)
ED_ENT_DISP_BANDEJA	(3, 20)
ED_SAL_DESVIO_3	(2, 20)
ED_SAL_DISP_BANDEJA	(5, 20)
ED_SAL_SCARA	(5, 15)
INICIO_ACUMULA	(1, 20)
SD_MOTOR_3	(7, 20)

Chart : [MAST - MODO_AUTO]



Comunicacion_MODBUS : [MAST]



**Etiquetas truncadas:**

Etiqueta	Posición(es)
'{192.168.0.15}'	(1, 4)
Escritura_peticion_alim_band	(1, 14)
info_lectura_alim_bandejas[1]	(1, 33)
pet_alimentador_band_A_ESCRIBIR	(1, 19)
tabla_gestion_lectura_BOOL	(1, 28)

ComunicacionPuertoSerie : [MAST]

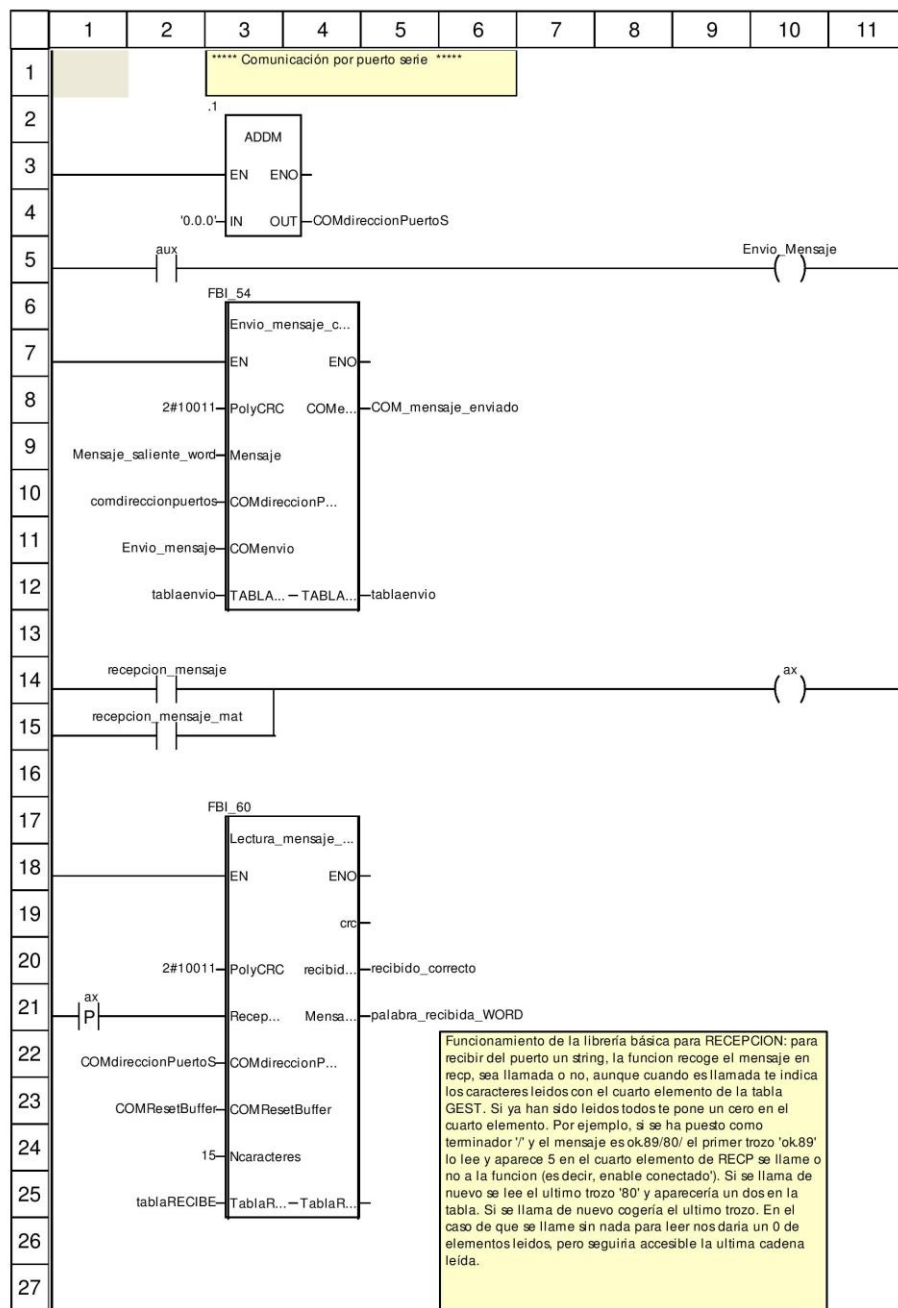
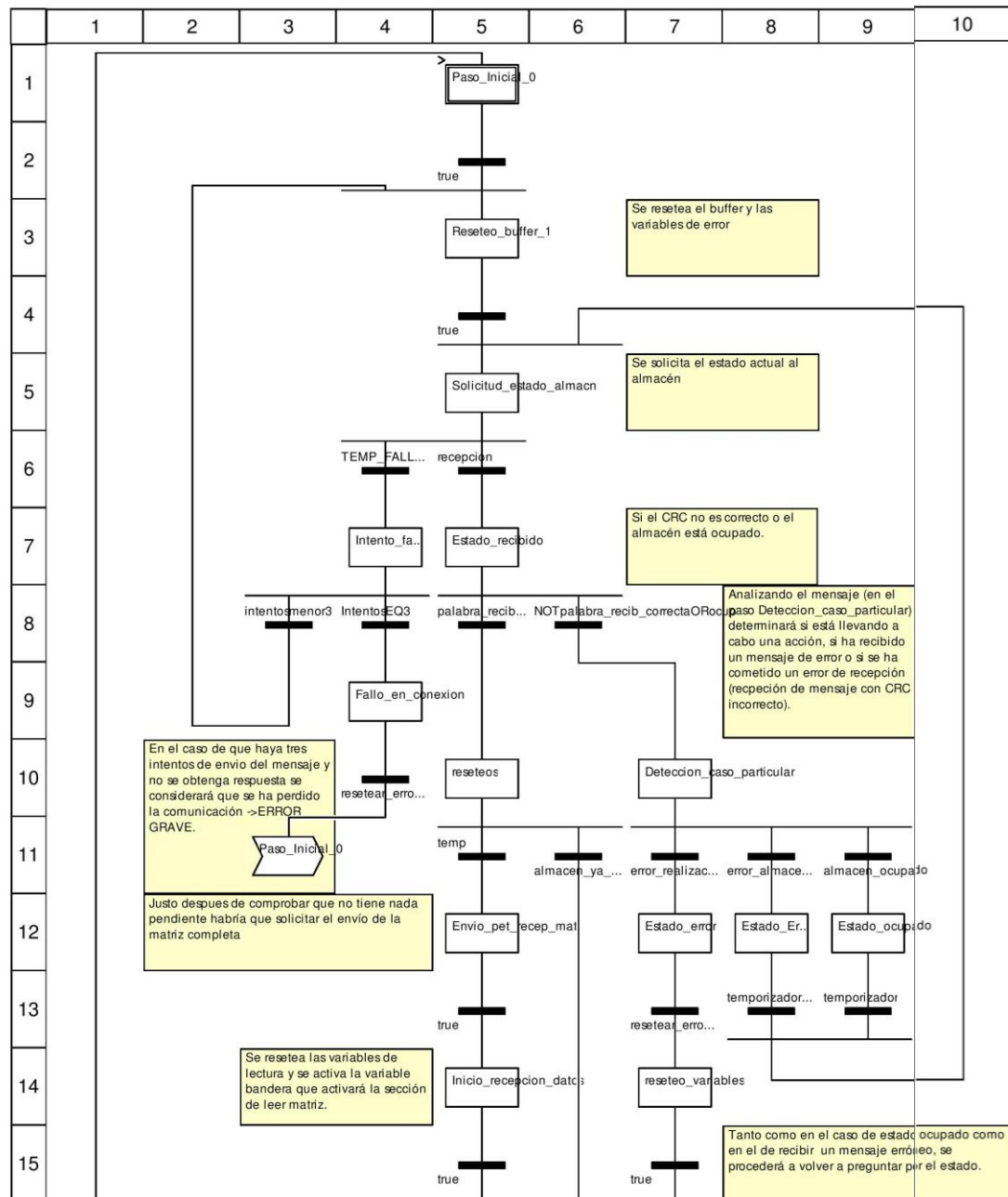


Chart : [MAST - MODO_OPERA_ALMACEN]



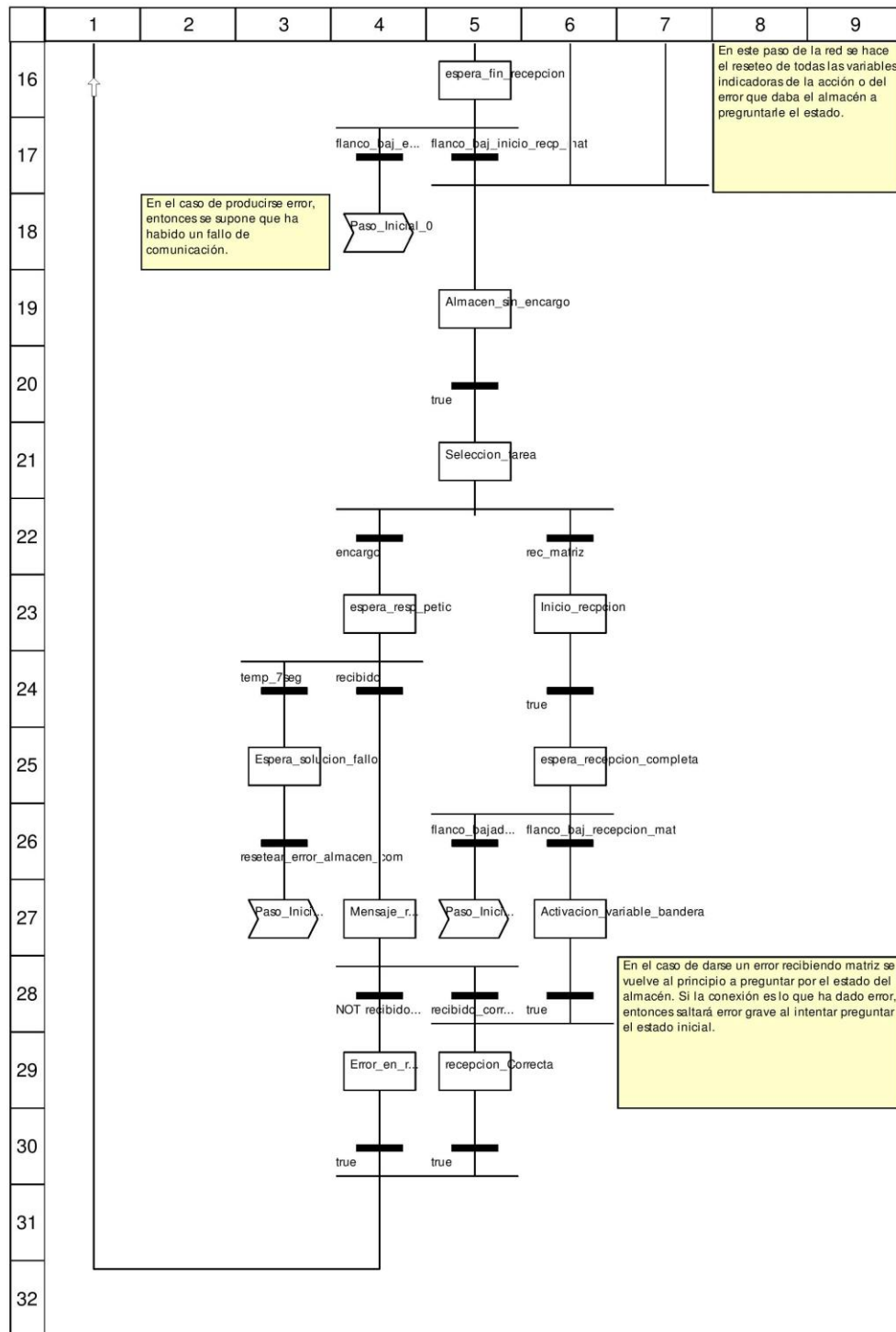
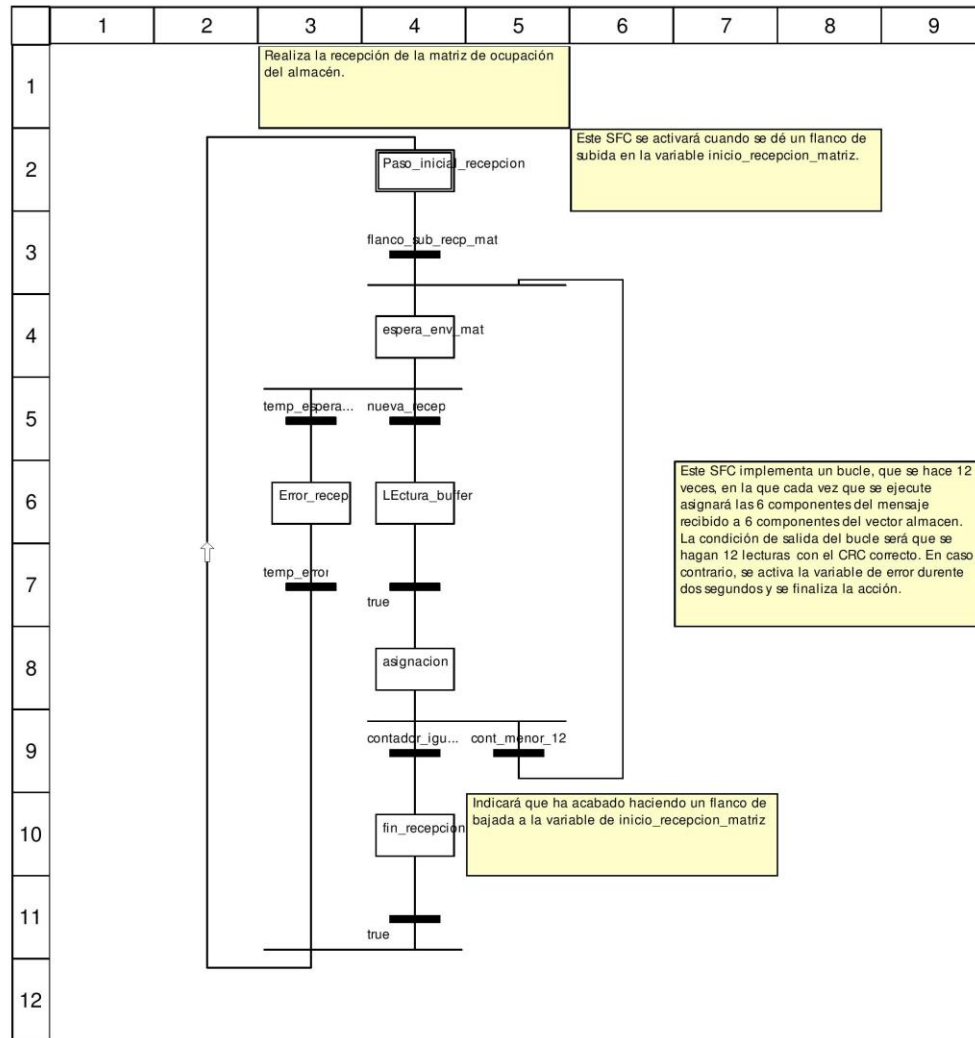


Chart : [MAST - lee_matriz_sfc]



SeguidorVisual : [MAST]

Propiedades específicas

Nombre de la condición	COND_SEG_BANDEJAS
1	10
20	30
40	50
60	70
80	90
100	110
1	
2	
3	
4	(* ***** *)
5	(* ACTUALIZACIÓN DE LA INFORMACIÓN DE CADA TRAMO *)
6	(* ***** *)
7	
8	(* A continuación se irá actualizando la info de los tramos (si el tramo
9	está lleno y qué hay en ese caso*)
10	
11	(* TRAMO 1: FORMADO POR LOS SUBTRAMOS 1,2,3 Y 4 *)
12	
13	IF NOT PALET_ELEV_PUSH_4_ANT AND ED_PALET_ELEV_PUSH_4 THEN
14	Tramo1[1].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
15	Tramo1[1].palet:=tramo4[4].palet; (*Indica si la bandeja lleva palet*)
16	Tramo1[1].procesado:=tramo4[4].procesado; (*Indica si el palet está procesado*)
17	Tramo1[1].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
18	
19	Tramo4[4].Bandeja:=0;
20	Tramo4[4].Palet:=0;
21	Tramo4[4].Procesado:=0;
22	END_IF;
23	
24	
25	IF NOT ED_ENT_ALMACEN AND ENT_ALMACEN_ANT THEN
26	Tramo1[2].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
27	Tramo1[2].palet:=tramo1[1].palet; (*Indica si la bandeja lleva palet*)
28	Tramo1[2].procesado:=tramo1[1].procesado; (*Indica si el palet está procesado*)
29	Tramo1[2].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
30	
31	Tramo1[1].Bandeja:=0;
32	Tramo1[1].Palet:=0;
33	Tramo1[1].Procesado:=0;
34	END_IF;
35	
36	
37	IF NOT ED_ENT_DESVIO_5 AND ENT_DESVIO_5_ANT THEN
38	Tramo1[3].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
39	Tramo1[3].palet:=tramo1[2].palet; (*Indica si la bandeja lleva palet*)
40	Tramo1[3].procesado:=tramo1[2].procesado; (*Indica si el palet está procesado*)
41	Tramo1[3].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
42	
43	Tramo1[2].Bandeja:=0;
44	Tramo1[2].Palet:=0;
45	Tramo1[2].Procesado:=0;
46	END_IF;
47	
48	IF NOT ED_ENT_DESVIO_1 AND ENT_DESVIO_1_ANT THEN
49	Tramo1[4].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
50	Tramo1[4].palet:=tramo1[3].palet; (*Indica si la bandeja lleva palet*)
51	Tramo1[4].procesado:=tramo1[3].procesado; (*Indica si el palet está procesado*)
52	Tramo1[4].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
53	
54	Tramo1[3].Bandeja:=0;
55	Tramo1[3].Palet:=0;
56	Tramo1[3].Procesado:=0;
57	END_IF;
58	
59	(* TRAMO 2: DIVIDIDO POR LOS SUBTRAMOS 1, 2 Y 3 *)
60	
61	IF NOT ED_PALET_ELEV_1 AND PALET_ELEV_1_ANT THEN
62	Tramo2[1].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
63	Tramo2[1].palet:=tramo1[4].palet; (*Indica si la bandeja lleva palet*)
64	Tramo2[1].procesado:=tramo1[4].procesado; (*Indica si el palet está procesado*)
65	Tramo2[1].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
66	
67	Tramo1[4].Bandeja:=0;
68	Tramo1[4].Palet:=0;
69	Tramo1[4].Procesado:=0;
70	END_IF;
71	
72	IF NOT ED_SAL_DESVIO_1 AND SAL_DESVIO_1_ANT THEN
73	Tramo2[2].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
74	Tramo2[2].palet:=tramo2[1].palet; (*Indica si la bandeja lleva palet*)
75	Tramo2[2].procesado:=tramo2[1].procesado; (*Indica si el palet está procesado*)
76	Tramo2[2].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
77	
78	Tramo2[1].Bandeja:=0;
79	Tramo2[1].Palet:=0;

SeguidorVisual

```

1|      10|      20|      30|      40|      50|      60|      70|      80|      90|     100|     110|
80  Tramo2[1].Procesado:=0;
81  END_IF;
82
83  IF NOT ED_ENT_DESVIO_2 AND ENT_DESVIO_2_ANT THEN
84  Tramo2[3].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
85  Tramo2[3].palet:=tramo2[2].palet; (*Indica si la bandeja lleva palet*)
86  Tramo2[3].procesado:=tramo2[2].procesado; (*Indica si el palet está procesado*)
87  Tramo2[3].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
88
89  Tramo2[2].Bandeja:=0;
90  Tramo2[2].Palet:=0;
91  Tramo2[2].Procesado:=0;
92  END_IF;
93
94  (* TRAMO 3: DIVIDIDO POR LOS SUBTRAMOS 1, 2 Y 3*)
95
96  IF NOT PALET_ELEV_PUSH_2_ANT AND ED_PALET_ELEV_PUSH_2 THEN
97  Tramo3[1].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
98  Tramo3[1].palet:=tramo2[3].palet; (*Indica si la bandeja lleva palet*)
99  Tramo3[1].procesado:=tramo2[3].procesado; (*Indica si el palet está procesado*)
100 Tramo3[1].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
101
102 Tramo2[3].Bandeja:=0;
103 Tramo2[3].Palet:=0;
104 Tramo2[3].Procesado:=0;
105 END_IF;
106
107 IF ENT_SCARA_ANT AND not ED_ENT_SCARA THEN
108 Tramo3[2].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
109 Tramo3[2].palet:=tramo3[1].palet; (*Indica si la bandeja lleva palet*)
110 Tramo3[2].procesado:=tramo3[1].procesado; (*Indica si el palet está procesado*)
111 Tramo3[2].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
112
113 Tramo3[1].Bandeja:=0;
114 Tramo3[1].Palet:=0;
115 Tramo3[1].Procesado:=0;
116 END_IF;
117
118 IF NOT ENT_DESVIO_3_ANT AND ED_ENT_DESVIO_3 THEN
119 Tramo3[3].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
120 Tramo3[3].palet:=tramo3[2].palet; (*Indica si la bandeja lleva palet*)
121 Tramo3[3].procesado:=tramo3[2].procesado; (*Indica si el palet está procesado*)
122 Tramo3[3].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
123
124 Tramo3[2].Bandeja:=0;
125 Tramo3[2].Palet:=0;
126 Tramo3[2].Procesado:=0;
127 END_IF;
128
129 (* TRAMO 4: DIVIDIDO POR LOS SUBTRAMOS 1,2,3 y 4 *)
130
131 IF NOT PALET_ELEV_3_ANT AND ED_PALET_ELEV_3 THEN
132 Tramo4[1].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
133 Tramo4[1].palet:=tramo3[3].palet; (*Indica si la bandeja lleva palet*)
134 Tramo4[1].procesado:=tramo3[3].procesado; (*Indica si el palet está procesado*)
135 Tramo4[1].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
136
137 Tramo3[3].Bandeja:=0;
138 Tramo3[3].Palet:=0;
139 Tramo3[3].Procesado:=0;
140 END_IF;
141
142 IF NOT ED_SAL_DESVIO_3 AND SAL_DESVIO_3_ANT THEN
143 Tramo4[2].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
144 Tramo4[2].palet:=tramo4[1].palet; (*Indica si la bandeja lleva palet*)
145 Tramo4[2].procesado:=tramo4[1].procesado; (*Indica si el palet está procesado*)
146 Tramo4[2].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
147
148 Tramo4[1].Bandeja:=0;
149 Tramo4[1].Palet:=0;
150 Tramo4[1].Procesado:=0;
151 END_IF;
152
153 IF not ED_ENT_DISP_BANDEJA AND ENT_DISPENSADOR_ant THEN
154 Tramo4[3].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
155 Tramo4[3].palet:=tramo4[2].palet; (*Indica si la bandeja lleva palet*)
156 Tramo4[3].procesado:=tramo4[2].procesado; (*Indica si el palet está procesado*)
157 Tramo4[3].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
158
159 Tramo4[2].Bandeja:=0;
160 Tramo4[2].Palet:=0;
161 Tramo4[2].Procesado:=0;
162 END_IF;
163
164 IF not ED_ENT_DESVIO_4 AND ENT_DESVIO_4_ant THEN
165 Tramo4[4].Bandeja:=1; (*Indica si hay bandeja en el tramo *)
166 Tramo4[4].palet:=tramo4[3].palet; (*Indica si la bandeja lleva palet*)
167 Tramo4[4].procesado:=tramo4[3].procesado; (*Indica si el palet está procesado*)
168 Tramo4[4].tiempo:=0; (*Indica el tiempo que lleva en el proceso la pieza*)
169
170 Tramo4[3].Bandeja:=0;
171
172
173 Tramo4[3].Bandeja:=0;

```

SeguidorVisual

```

1|      10|      20|      30|      40|      50|      60|      70|      80|      90|     100|     110|
174  Tramo4[3].Palet:=0;
175  Tramo4[3].Procesado:=0;
176  END_IF;
177
178  (* ***** *)
179
180
181
182  (* ***** *)
183  (* CONTADORES DE TIEMPO para cada sección *)
184  (* ***** *)
185
186  (* en el caso de superar un tiempo límite, se considerará perdida la
187    bandeja en cuestión *)
188
189  IF Tramo1[1].bandeja=1 and not ED_ENT_ALMACEN THEN
190
191      IF salida_pulsos THEN
192          Tramo1[1].tiempo:=Tramo1[1].tiempo+1;
193      END_IF;
194      (*Si está en el tramo un tiempo máximo, se considerará que
195      la bandeja ha sido quitada por alguien *)
196      IF Tramo1[1].tiempo=100 THEN
197          Tramo1[1].bandeja:=0;
198          Tramo1[1].palet:=0;
199          Tramo1[1].procesado:=0;
200      END_IF;
201  END_IF;
202
203
204  IF Tramo1[2].bandeja=1 and not ED_ENT_DESVIO_5 THEN
205      IF salida_pulsos THEN
206          Tramo1[2].tiempo:=Tramo1[2].tiempo+1;
207      END_IF;
208      (*Si está en el tramo un tiempo máximo, se considerará que
209      la bandeja ha sido quitada por alguien *)
210      IF Tramo1[2].tiempo=100 THEN
211          Tramo1[2].bandeja:=0;
212          Tramo1[2].palet:=0;
213          Tramo1[2].procesado:=0;
214      END_IF;
215  END_IF;
216
217
218  IF Tramo1[3].bandeja=1 and not ED_ENT_DESVIO_1 THEN
219      IF salida_pulsos THEN
220          Tramo1[3].tiempo:=Tramo1[3].tiempo+1;
221      END_IF;
222      (*Si está en el tramo un tiempo máximo, se considerará que
223      la bandeja ha sido quitada por alguien *)
224      IF Tramo1[3].tiempo=100 THEN
225          Tramo1[3].bandeja:=0;
226          Tramo1[3].palet:=0;
227          Tramo1[3].procesado:=0;
228      END_IF;
229  END_IF;
230
231  (*Tendría que poner alguna condición para distinguir cuando está esperando a que se quede libre el otro tramo
231>>*)
232  IF Tramo1[4].bandeja=1 THEN
233      IF salida_pulsos THEN
234          Tramo1[4].tiempo:=Tramo1[4].tiempo+1;
235      END_IF;
236      (*Si está en el tramo un tiempo máximo, se considerará que
237      la bandeja ha sido quitada por alguien *)
238      IF Tramo1[4].tiempo=100 THEN
239          Tramo1[4].bandeja:=0;
240          Tramo1[4].palet:=0;
241          Tramo1[4].procesado:=0;
242      END_IF;
243  END_IF;
244
245
246
247  IF Tramo2[1].bandeja=1 THEN
248      IF salida_pulsos THEN
249          Tramo2[1].tiempo:=Tramo2[1].tiempo+1;
250      END_IF;
251      (*Si está en el tramo un tiempo máximo, se considerará que
252      la bandeja ha sido quitada por alguien *)
253      IF Tramo2[1].tiempo=160 THEN
254          Tramo2[1].bandeja:=0;
255          Tramo2[1].palet:=0;
256          Tramo2[1].procesado:=0;
257      END_IF;
258  END_IF;
259
260
261  IF Tramo2[2].bandeja=1 and not ED_ENT_DESVIO_2 THEN
262      IF salida_pulsos THEN
263          Tramo2[2].tiempo:=Tramo2[2].tiempo+1;
264      END_IF;
265      (*Si está en el tramo un tiempo máximo, se considerará que
266      la bandeja ha sido quitada por alguien *)

```

SeguidorVisual

```

1|      10|      20|      30|      40|      50|      60|      70|      80|      90|      100|      110|
267      IF Tramo2[2].tiempo=100 THEN
268      Tramo2[2].bandeja:=0;
269      Tramo2[2].palet:=0;
270      Tramo2[2].procesado:=0;
271      END_IF;
272 END_IF;
273
274 IF Tramo2[3].bandeja=1 and ED_PALET_ELEV_PUSH_2 THEN
275 IF salida_pulsos THEN
276 Tramo2[3].tiempo:=Tramo2[3].tiempo+1;
277 END_IF;
278 (*Si está en el tramo un tiempo máximo, se considerará que
279 la bandeja ha sido quitada por alguien *)
280 IF Tramo2[3].tiempo=100 THEN
281 Tramo2[3].bandeja:=0;
282 Tramo2[3].palet:=0;
283 Tramo2[3].procesado:=0;
284 END_IF;
285 END_IF;
286
287 IF Tramo3[1].bandeja=1 THEN
288
289
290 IF salida_pulsos and not ED_ENT_SCARA THEN
291 Tramo3[1].tiempo:=Tramo3[1].tiempo+1;
292 END_IF;
293 (*Si está en el tramo un tiempo máximo, se considerará que
294 la bandeja ha sido quitada por alguien *)
295 IF Tramo3[1].tiempo=100 THEN
296 Tramo3[1].bandeja:=0;
297 Tramo3[1].palet:=0;
298 Tramo3[1].procesado:=0;
299 END_IF;
300
301
302 END_IF;
303
304 IF Tramo3[2].bandeja=1 and not ED_ENT_DESVIO_3 THEN
305
306
307 IF salida_pulsos THEN
308 Tramo3[2].tiempo:=Tramo3[2].tiempo+1;
309 END_IF;
310 (*Si está en el tramo un tiempo máximo, se considerará que
311 la bandeja ha sido quitada por alguien *)
312 IF Tramo3[2].tiempo=100 THEN
313 Tramo3[2].bandeja:=0;
314 Tramo3[2].palet:=0;
315 Tramo3[2].procesado:=0;
316 END_IF;
317 END_IF;
318
319 IF Tramo3[3].bandeja=1 THEN
320 IF salida_pulsos THEN
321 Tramo3[3].tiempo:=Tramo3[3].tiempo+1;
322 END_IF;
323 (*Si está en el tramo un tiempo máximo, se considerará que
324 la bandeja ha sido quitada por alguien *)
325 IF Tramo3[3].tiempo=100 THEN
326 Tramo3[3].bandeja:=0;
327 Tramo3[3].palet:=0;
328 Tramo3[3].procesado:=0;
329 END_IF;
330 END_IF;
331
332 IF Tramo4[1].bandeja=1 THEN
333 IF salida_pulsos THEN
334 Tramo4[1].tiempo:=Tramo4[1].tiempo+1;
335 END_IF;
336 (*Si está en el tramo un tiempo máximo, se considerará que
337 la bandeja ha sido quitada por alguien *)
338 IF Tramo4[1].tiempo=160 THEN
339 Tramo4[1].bandeja:=0;
340 Tramo4[1].palet:=0;
341 Tramo4[1].procesado:=0;
342 END_IF;
343 END_IF;
344
345 IF Tramo4[2].bandeja=1 and not ED_ENT_DISP_BANDEJA THEN
346 IF salida_pulsos THEN
347 Tramo4[2].tiempo:=Tramo4[2].tiempo+1;
348 END_IF;
349 (*Si está en el tramo un tiempo máximo, se considerará que
350 la bandeja ha sido quitada por alguien *)
351 IF Tramo4[2].tiempo=100 THEN
352 Tramo4[2].bandeja:=0;
353 Tramo4[2].palet:=0;
354 Tramo4[2].procesado:=0;
355 END_IF;
356 END_IF;
357
358 IF Tramo4[3].bandeja=1 THEN
359 IF salida_pulsos and not ED_ENT_DESVIO_4 THEN
360 Tramo4[3].tiempo:=Tramo4[3].tiempo+1;

```


SeguidorVisual

```

1|      10|      20|      30|      40|      50|      60|      70|      80|      90|     100|     110|
361      IF ALMACENANDO_bandeja AND Tramo4[3].tiempo=26 THEN
362      Tramo4[3].tiempo:=25;
363      end_if;
364      END_IF;
365      (*Si se detecta palet y se sabe que no está el palet procesado*)
366      IF PLC2_s11_barrido=1 AND Tramo4[3].procesado=0 THEN
367      Tramo4[3].palet:=1;
368      Tramo4[3].bandeja:=1;
369      END_IF;
370      (*Si está en el tramo un tiempo máximo, se considerará que
371      la bandeja ha sido quitada por alguien *)
372      IF Tramo4[3].tiempo=160 and not ED_ENT_DESVIO_4 THEN
373      Tramo4[3].bandeja:=0;
374      Tramo4[3].palet:=0;
375      Tramo4[3].procesado:=0;
376      END_IF;
377      end_if;
378
379      IF Tramo4[4].Bandeja=1 and ED_PALET_ELEV_PUSH_4 THEN
380      IF salida_pulsos THEN
381      Tramo4[4].tiempo:=Tramo4[4].tiempo+1;
382      END_IF;
383      (*Si está en el tramo un tiempo máximo, se considerará que
384      la bandeja ha sido quitada por alguien *)
385      IF Tramo4[4].tiempo=100 THEN
386      Tramo4[4].bandeja:=0;
387      Tramo4[4].palet:=0;
388      Tramo4[4].procesado:=0;
389      END_IF;
390      END_IF;
391
392      (* ***** *)
393      (* Actualización de variables para la detección de flancos *)
394      (* ***** *)
395      sacando_bandeja_ant:=sacando_bandeja;
396      almacenando_bandeja_ant:=almacenando_bandeja;
397      PALET_ELEV_PUSH_4_ANT:=ED_PALET_ELEV_PUSH_4;
398      PALET_ELEV_PUSH_2_ANT:=ED_PALET_ELEV_PUSH_2;
399      ENT_ALMACEN_ANT:=ED_ENT_ALMACEN;
400      ENT_DESVIO_1_ANT:=ED_ENT_DESVIO_1;
401      ENT_DESVIO_2_ANT:=ED_ENT_DESVIO_2;
402      ENT_DESVIO_3_ANT:=ED_ENT_DESVIO_3;
403      ENT_DESVIO_4_ANT:=ED_ENT_DESVIO_4;
404      ENT_DESVIO_5_ANT:=ED_ENT_DESVIO_5;
405      ENT_DISP_BANDEJA_ANT:=ED_ENT_DISP_BANDEJA;
406      ENT_DISPENSADOR_ANT:=ED_ENT_DISP_BANDEJA;
407      ENT_SCARA_ANT:=ED_ENT_SCARA;
408      PALET_ELEV_1_ANT:=ED_PALET_ELEV_1;
409      PALET_ELEV_3_ANT:=ED_PALET_ELEV_3;
410      SAL_DESVIO_1_ANT:=ED_SAL_DESVIO_1;
411      SAL_DESVIO_3_ANT:=ED_SAL_DESVIO_3;
412
413
414
415      (* ***** *)
416      (* Cálculo del número de piezas actual en la instalación *)
417      (* ***** *)
418
419      (* Se usan bucles for para recorrer los tramos y realizar las sumas. Con el fin
420      de no realizar cada ciclo estas operaciones, se hará sólo cada 200milisegundos (a la
421      señal del generador de pulsos *)
422
423
424
425      Band_libres_circ:=0;
426      Band_palet_sin_procesar:=0;
427      Band_palet_procesado:=0;
428      FOR cont_aux:=1 TO 4 BY 1 DO
429      if (tramo1[cont_aux].bandeja)AND(tramo1[cont_aux].palet)AND(tramo1[cont_aux].procesado) then
430      band_palet_procesado:=band_palet_procesado+1;
431      end_if;
432      if (tramo1[cont_aux].bandeja)AND(tramo1[cont_aux].palet)AND(not tramo1[cont_aux].procesado) then
433      band_palet_sin_procesar:=band_palet_sin_procesar+1;
434      end_if;
435      if (tramo1[cont_aux].bandeja)AND(not tramo1[cont_aux].palet) then
436      band_libres_circ:=band_libres_circ+1;
437      end_if;
438      END_FOR;
439
440      FOR cont_aux:=1 TO 3 BY 1 DO
441      if (tramo2[cont_aux].bandeja)AND(tramo2[cont_aux].palet)AND(tramo2[cont_aux].procesado) then
442      band_palet_procesado:=band_palet_procesado+1;
443      end_if;
444      if (tramo2[cont_aux].bandeja)AND(tramo2[cont_aux].palet)AND(not tramo2[cont_aux].procesado) then
445      band_palet_sin_procesar:=band_palet_sin_procesar+1;
446      end_if;
447      if (tramo2[cont_aux].bandeja)AND(not tramo2[cont_aux].palet) then
448      band_libres_circ:=band_libres_circ+1;
449      end_if;
450      END_FOR;
451
452      FOR cont_aux:=1 TO 3 BY 1 DO
453      if (tramo3[cont_aux].bandeja)AND(tramo3[cont_aux].palet)AND(tramo3[cont_aux].procesado) then
454

```

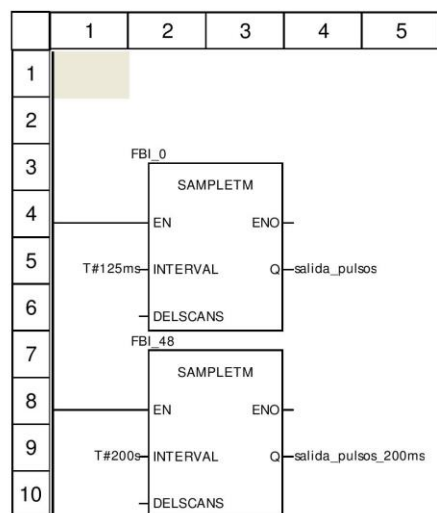
SeguidorVisual

```

1|      10|      20|      30|      40|      50|      60|      70|      80|      90|      100|      110|
455      band_palet_procesado:=band_palet_procesado+1;
456      end_if;
457      if (tramo3[cont_aux].bandeja)AND(tramo3[cont_aux].palet)AND(not tramo3[cont_aux].procesado) then
458      band_palet_sin_procesar:=band_palet_sin_procesar+1;
459      end_if;
460      if (tramo3[cont_aux].bandeja)AND( not tramo3[cont_aux].palet) then
461      band_libres_circ:=band_libres_circ+1;
462      end_if;
463      END_FOR;
464
465      FOR cont_aux:=1 TO 4 BY 1 DO
466      if (tramo4[cont_aux].bandeja)AND(tramo4[cont_aux].palet)AND(tramo4[cont_aux].procesado) then
467      band_palet_procesado:=band_palet_procesado+1;
468      end_if;
469      if (tramo4[cont_aux].bandeja)AND(tramo4[cont_aux].palet)AND(not tramo4[cont_aux].procesado) then
470      band_palet_sin_procesar:=band_palet_sin_procesar+1;
471      end_if;
472      if (tramo4[cont_aux].bandeja)AND(not tramo4[cont_aux].palet) then
473      band_libres_circ:=band_libres_circ+1;
474      end_if;
475      END_FOR;
476      band_circulacion:=Band_libres_circ+Band_palet_sin_procesar+Band_palet_procesado;
477

```

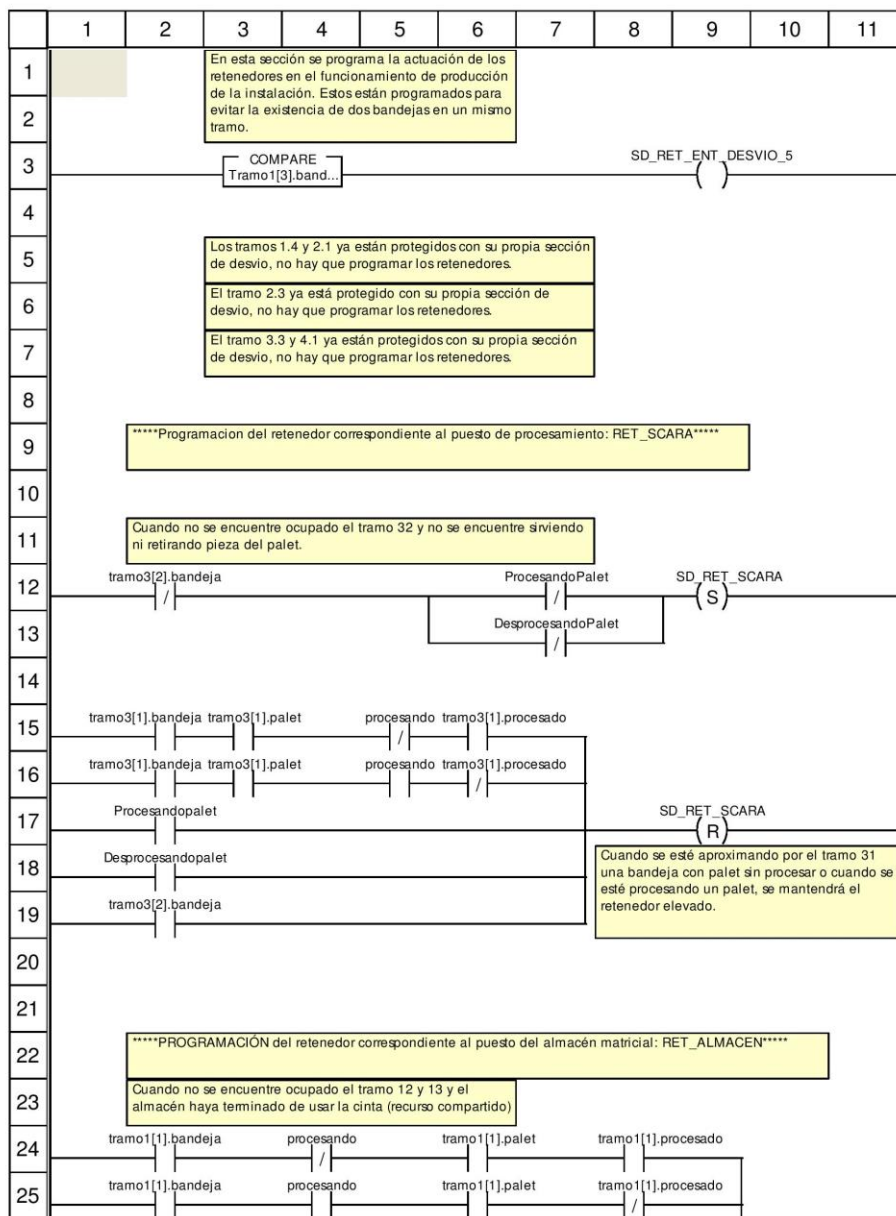
GeneradorPulsos : [MAST]

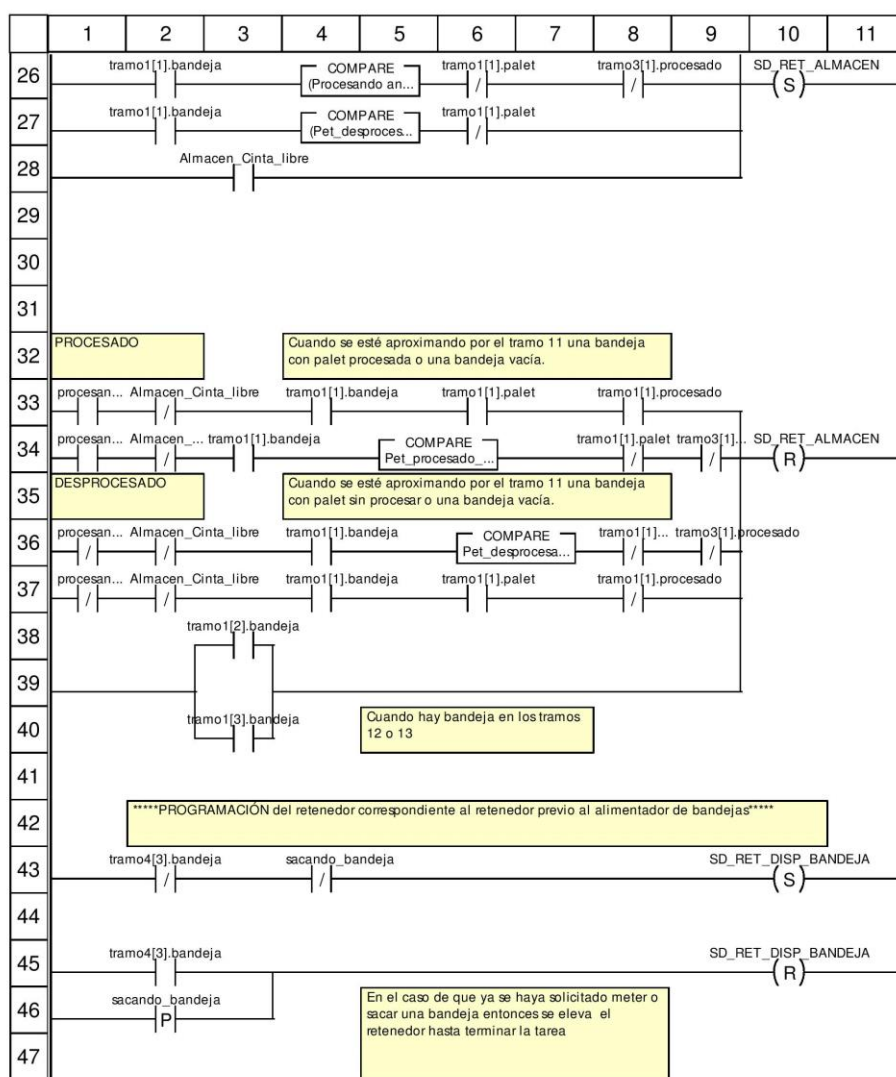


ProgramacionRetenedoresAUTO : [MAST]

Propiedades específicas

Nombre de la condición	COND_PROG_RET_AUTO
------------------------	--------------------





Etiquetas truncadas:

Etiqueta	Posición(es)
(Pet_desprocesado_piezas-Pet_en_proceso)=0 and not procesando	(4, 27)
(Procesando and (Pet_procesado_piezas-Pet_en_proceso=0))	(4, 26)
Almacen_Cinta_libre	(2, 34)
Pet_desprocesado_piezas-Pet_en_proceso>0	(6, 36)
Pet_procesado_piezas-Pet_en_proceso>0	(5, 34)
Tramo1[3].bandeja=0	(3, 3)
procesando	(1, 34) (1, 36) (1, 33)
tramo1[1].palet	(1, 37)
tramo3[1].procesado	(8, 36)
	(9, 34)

Gestion_peticiones_bandejas : [MAST]

Propiedades específicas

Nombre de la condición	cond_pet_bandejas
------------------------	-------------------

1| 10| 20| 30| 40| 50| 60| 70| 80| 90| 100| 110|

```

1
2  (*****
3  (*
4  (* GESTIÓN DE LAS PETICIONES DE EXTRAER O ALMACENAR BANDEJA EN EL ALIMENTADOR *)
5  (*
6  (*****
7
8
9  if TRAMO4[3].Bandeja=0 then
10
11      (* Se comprueba si es necesario Extraer una bandeja *)
12      IF not almacenando_bandeja and not sacando_bandeja and ( (PROCESANDO AND (Band_libres_circ<(Pet_proces
12>>ado_piezas-Pet_en_proceso))) or (NOT PROCESANDO AND (Band_libres_circ<(Pet_desprocesado_piezas-Pet_en_proceso
12>>))) ) AND Band_circulacion<Nmax_bandeja_instalacion AND
13  pet_bandejas_LEIDAS[3]>0 AND PLC2_Modo_remoto=1 then
14      (*Si la ejecución de procesos es menor que la de petición de procesado de las piezas*)
15
16      (*Se encarga servir una bandeja*)
17      pet_alimentador_band_A_ESCRIBIR[1]:=0;
18      pet_alimentador_band_A_ESCRIBIR[2]:=1;
19      (*Escribimos en el otro PLC*)
20      Escritura_peticion_alim_band:=true;
21
22      (* Se espera a que se quede marcada la petición en el otro PLC *)
23      if pet_alimentador_band_A_ESCRIBIR[1]=pet_bandejas_leidas[1] and pet_alimentador_band_A_ESCRIBIR[2]=pe
23>>t_bandejas_leidas[2] then
24          Escritura_peticion_alim_band:=false;
25          sacando_bandeja:=1; (* Cuando se está seguro que la petición ha llegado se marca sacando_bandeja *)
25>>)
26      end_if;
27
28      END_IF;
29
30      if sacando_bandeja and pet_bandejas_leidas[2]=0 and pet_band_serv_ant=1 then
31          (*cuando la petición de modo remoto en el alimentador disminuya
32          significará que ha extraído la bandeja*)
33          sacando_bandeja:=0;
34          NbandejasVirtual:=NbandejasVirtual+1; (*Será usada en el supervisor para luego detectar si alguien in
34>>troduce una bandeja*)
35          Tramo4[3].bandeja:=1;
36          Tramo4[3].palet:=0;
37          Tramo4[3].procesado:=0;
38          Tramo4[3].tiempo:=25;
39      end_if;
40
41      end_if;
42
43
44
45
46  if TRAMO4[2].Bandeja=1 and not tramo4[2].palet then
47      (* Se comprueba si es necesario almacenar una bandeja (Se supone que 9 es el máximo del almacén*)
48      IF not almacenando_bandeja and not sacando_bandeja and ( (PROCESANDO AND (Band_libres_circ>(Pet_proces
48>>ado_piezas-Pet_en_proceso))) or (NOT PROCESANDO AND (Band_libres_circ>(Pet_desprocesado_piezas-Pet_en_proceso)
48>>))) ) AND pet_bandejas_LEIDAS[3]<9 AND PLC2_modo_remoto=1 then
49
50      (*Se encarga de almacenar una bandeja*)
51      pet_alimentador_band_A_ESCRIBIR[1]:=1;
52      pet_alimentador_band_A_ESCRIBIR[2]:=0;
53      (*Se escribe en el otro PLC*)
54      Escritura_peticion_alim_band:=true;
55
56      (*Se comprueba que el encargo se escribe correctamente*)
57      if pet_alimentador_band_A_ESCRIBIR[1]=pet_bandejas_leidas[1] and pet_alimentador_band_A_ESCRIB
57>>IR[2]=pet_bandejas_leidas[2] then
58          Escritura_peticion_alim_band:=false;
59          (* Se marca que se está almacenando una bandeja *)
60          almacenando_bandeja:=1;
61      end_if;
62
63      end_if;
64      end_if;
65
66  if tramo4[3].bandeja=1 then
67      if almacenando_bandeja and pet_bandejas_leidas[1]=0 and pet_band_alm_ant=1 then
68          (*cuando la petición de modo remoto en el alimentador disminuya
69          significará que ha almacenado la bandeja*)
70          almacenando_bandeja:=0;
71          NbandejasVirtual:=NbandejasVirtual-1; (*Será usada en el supervisor para luego detectar si se introdu

```

Gestion_peticiones_bandejas

```

1|      10|      20|      30|      40|      50|      60|      70|      80|      90|     100|     110|
71>>ce manualmente una
72  bandeja*)
73      (*Se indica el final del proceso de almacenado y se marca la sección como vacía*)
74      Tramo4[3].bandeja:=0;
75      Tramo4[3].palet:=0;
76      Tramo4[3].procesado:=0;
77      END_IF;
78  end_if;
79
80
81
82
83  if not almacenando_bandeja and not sacando_bandeja and pet_bandejas_leidas[1]<>0 and pet_bandejas_leidas[2]<>0
83>> then
84      pet_alimentador_band_A_ESCRIBIR[1]:=0;
85      pet_alimentador_band_A_ESCRIBIR[2]:=0;
86      (*Se escribe en el otro PLC*)
87      Escritura_peticion_alim_band:=true;
88
89  end_if;
90
91
92  (*****
93  (*)
94  (*)      GESTIÓN DEL PROCESADO DE PIEZAS EN EL PUESTO DEL SCARA
95  (*)
96  (*****
97
98
99
100
101  IF Tramo3[1].bandeja=1 THEN
102
103      (* PROCESADO DE BANDEJAS *)
104
105      (* Si en este tramo se tiene una bandeja con palet no procesado y se está procesando entonces
106      se activa el retenedor en la seccion LD de retenedores *)
107
108      (* Si se tiene una pieza con palet no procesado entonces se procesa *)
109      IF Tramo3[1].palet=1 AND Tramo3[1].procesado=0 AND ED_ENT_SCARA AND Pet_en_Proceso>0 AND PROCESANDO then
110      ProcesandoPalet:=1;
111      inicio_temp_SCARA:=1;
112      end_if;
113
114      if ProcesandoPalet AND fin_temp_SCARA AND PROCESANDO then
115      ProcesandoPalet:=0;
116      Tramo3[1].procesado:=1;
117      inicio_temp_sCARA:=0;
118      end_if;
119
120      (* DESPROCESADO DE BANDEJAS *)
121
122      IF Tramo3[1].palet=1 AND Tramo3[1].procesado=1 AND ED_ENT_SCARA AND Pet_en_Proceso>0 AND NOT PROCESANDO t
122>>hen
123      DesprocesandoPalet:=1;
124      inicio_temp_SCARA:=1;
125      end_if;
126
127      if DesprocesandoPalet AND fin_temp_SCARA AND NOT PROCESANDO then
128      DesprocesandoPalet:=0;
129      Tramo3[1].procesado:=0;
130      inicio_temp_sCARA:=0;
131      end_if;
132
133  end_if;
134
135
136  (*****
137  (*)
138  (*)      GESTIÓN DE LAS PETICIONES DE EXTRAER O ALMACENAR PALET EN EL ALMACÉN MATRICIAL
139  (*)
140  (*****
141
142
143  IF Tramo1[1].bandeja=1 THEN
144
145      (* CASO DEL MODO PROCESANDO *)
146
147      (* Si en el tramo 11 hay una bandeja con palet ya procesado cuando se está en modo procesando y
148      el almacén está en espera de pedido se almacena *)
149      IF PROCESANDO AND ED_ENT_ALMACEN AND Tramo1[1].palet AND Tramo1[1].procesado and almacen_esperando_petic
149>>ion THEN
150      mete_pieza_tratada:=1;
151      END_IF;
152
153
154      IF PROCESANDO and not A3_PARADA_PEDIDA AND not Tramo1[1].palet AND not Tramo1[1].procesado and almacen_e
154>>sperando_peticion and (Pet_procesado_piezas-Pet_en_proceso)>0 THEN
155      saca_pieza_sin_tratar:=1;
156      END_IF;
157
158      (* CASO DEL MODO DESPROCESANDO (=NOT PROCESANDO) *)
159      (* En este caso se almacenan las bandejas sin procesar y se extraen las procesadas*)
160

```

Gestion_peticiones_bandejas

```

161      IF NOT PROCESANDO AND ED_ENT_ALMACEN AND Tramol[1].palet AND not Tramol[1].procesado AND almacen_esperan
161>>do_peticion THEN
162      mete_pieza_sin_tratar:=1;
163      END_IF;
164
165
166      IF NOT PROCESANDO and not A3_PARADA_PEDIDA AND not Tramol[1].palet and almacen_esperando_peticion and (Pe
166>>t_desprocesado_piezas-Pet_en_proceso)>0 THEN
167      saca_pieza_tratada:=1;
168      END_IF;
169
170
171      (* DESACTIVACIÓN DE VARIABLES DE PETICIÓN DEL ALMACÉN *)
172
173
174      (*Cuando se detecte el inicio de la tarea a realizar se desactiva la variable de peticion *)
175      if sacando_pieza_sin_tratar=1 then
176      saca_pieza_sin_tratar:=0;
177      end_if;
178
179      if metiendo_pieza_tratada=1 then
180      mete_pieza_tratada:=0;
181      end_if;
182
183      if sacando_pieza_tratada=1 then
184      saca_pieza_tratada:=0;
185      end_if;
186
187      if metiendo_pieza_sin_tratar=1 then
188      mete_pieza_sin_tratar:=0;
189      end_if;
190
191
192      (* Cuando deja de trabajar el almacén en la zona se actualiza el tramo
193      y las variables de procesado *)
194      IF (Almacen_Cinta_libre and not Almacen_Cinta_libre_ant) then
195      (*Cuando se almacene la pieza se considera que ya se ha procesado la pieza*)
196
197      (* CASO DEL MODO PROCESANDO *)
198
199      if metiendo_pieza_tratada=1 then
200      Tramol[1].palet:=0;
201      Tramol[1].procesado:=0;
202      Pet_procesado_piezas:=Pet_procesado_piezas-1;
203      Pet_en_proceso:=Pet_en_proceso-1;
204      end_if;
205
206      (*Cuando se extraiga la pieza se aumenta la variable de peticiones en proceso*)
207      if sacando_pieza_sin_tratar=1 or (sacando_pieza_sin_tratar_ant and not sacando_pieza_sin_tratar) then
208      Tramol[1].palet:=1;
209      Tramol[1].procesado:=0;
210      Pet_en_proceso:=Pet_en_proceso+1;
211      end_if;
212      (* CASO DEL MODO DESPROCESANDO *)
213      if metiendo_pieza_sin_tratar=1 then
214      Tramol[1].palet:=0;
215      Tramol[1].procesado:=0;
216      Pet_desprocesado_piezas:=Pet_desprocesado_piezas-1;
217      Pet_en_proceso:=Pet_en_proceso-1;
218      end_if;
219
220      (*Cuando se saque la pieza se aumenta la variable de peticiones en proceso*)
221      if sacando_pieza_tratada=1 or (sacando_pieza_tratada_ant and not sacando_pieza_tratada) then
222      Tramol[1].palet:=1;
223      Tramol[1].procesado:=1;
224      Pet_en_proceso:=Pet_en_proceso+1;
225      end_if;
226
227      END_IF;
228
229  end_if;
230
231  (*****
232  (*)
233  (*)          GESTIÓN AUTOMÁTICA DEL NÚMERO DE PETICIONES DEL SISTEMA          (*)
234  (*)
235  (*****
236
237  IF Seleccion_peticion_automatica then
238
239  (* Se calcula PRIMERO LOS NÚMEROS DE PIEZAS DE CADA TIPO EN EL ALMACÉN-> ahora se realiza
240  cada vez que se reciba la matriz de ocupación del almacén *)
241
242  if not A4_PARADA_OBTENIDA and not A3_PARADA_PEDIDA then
243  (* En el caso de que se tenga una solicitud de parada en el modo automático o se esté
244  en parada, se cancela el cálculo de las peticiones automáticas*)
245
246  (* Asignación de las peticiones *)
247
248  if palet_procesados<24 AND PROCESANDO=1 then
249  Pet_procesado_piezas:=24-palet_procesados;
250  Pet_desprocesado_piezas:=0;
251  END_IF;
252

```

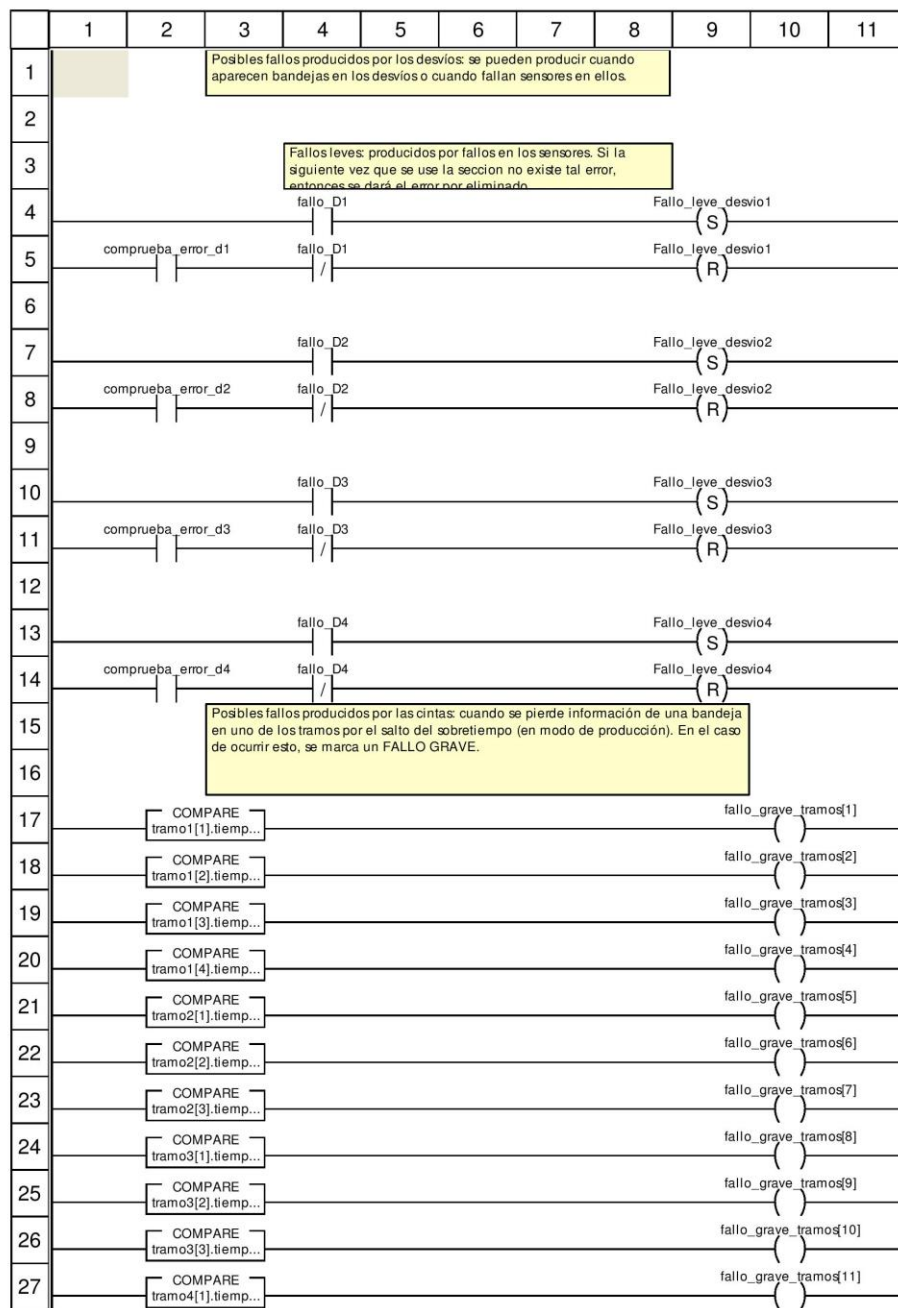

Gestion_peticiones_bandejas

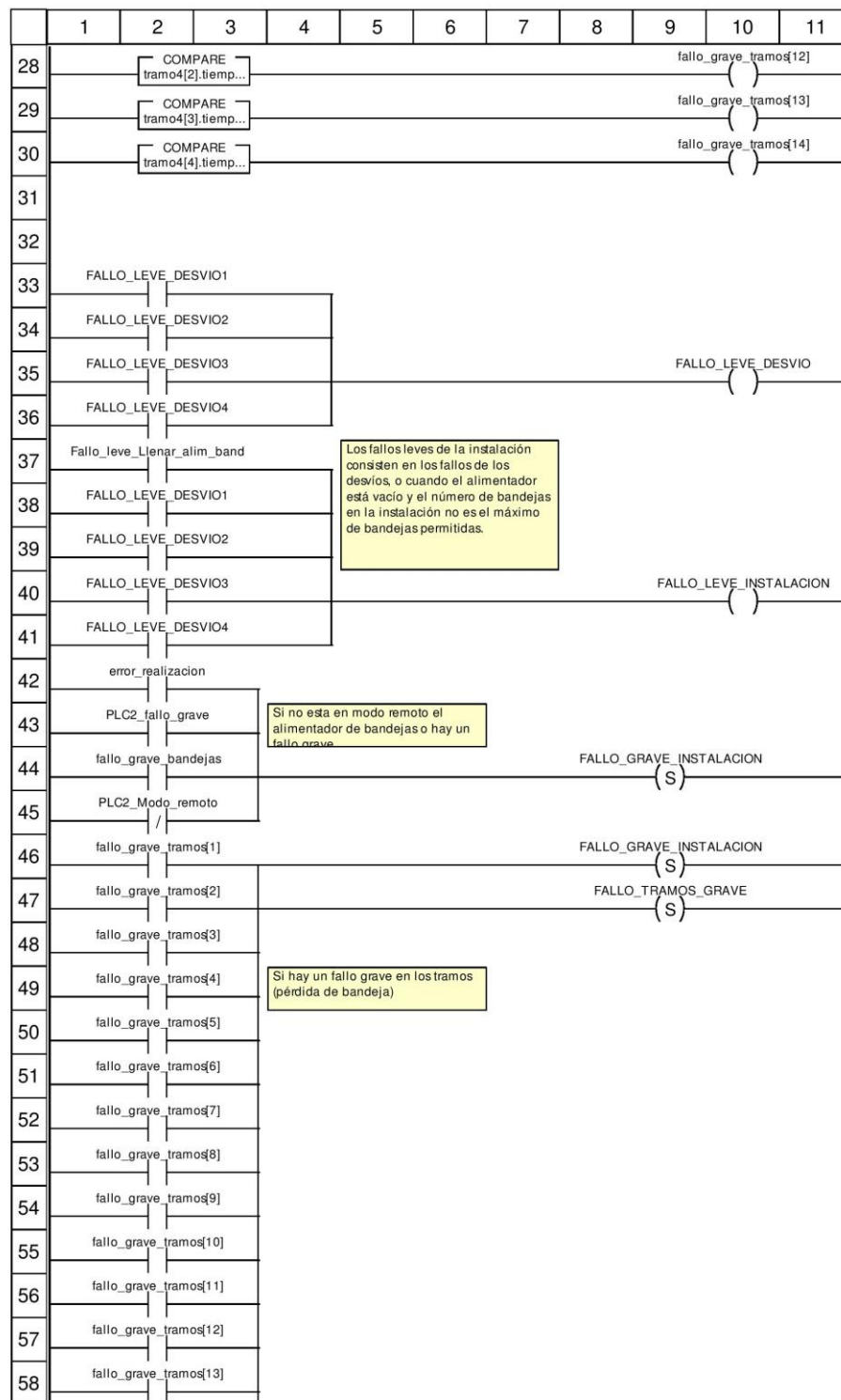
```

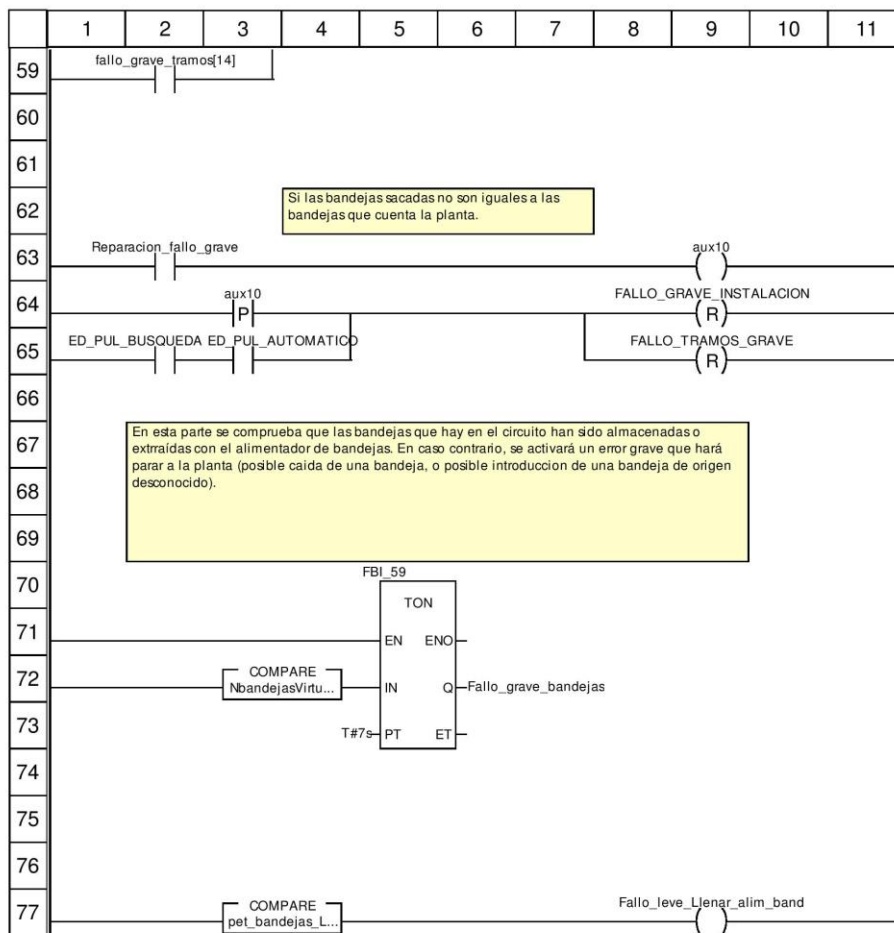
1|      10|      20|      30|      40|      50|      60|      70|      80|      90|     100|     110|
253      if palet_procesados>0 AND PROCESANDO=0 then
254          if (36-Palet_sin_procesar)>palet_procesados then
255              Pet_desprocesado_piezas:=palet_procesados;
256          else
257              Pet_desprocesado_piezas:=36-Palet_sin_procesar;
258          end_if;
259
260          Pet_procesado_piezas:=0;
261
262          END_IF;
263
264      end_IF;
265
266      if palet_procesados=24 AND PROCESANDO=1 then
267          PROCESANDO:=0;
268      END_IF;
269
270      if palet_procesados=0 AND PROCESANDO=0 then
271          PROCESANDO:=1;
272      END_IF;
273
274      (* CASO DE PETICIONES MANUALES *)
275      (* En esta parte se limitan las peticiones introducidas en modo manual a las limitaciones
276      de ese momento en la instalación (número de huecos y de piezas libres)      *)
277
278      elsif PROCESANDO then
279
280          (* Si se tienen palets a procesar y huecos en la parte de procesados *)
281
282          if(24-Palet_procesados)>0 and (palet_sin_procesar)>0 THEN
283
284              (*Se estudia la casuística posible*)
285              if (24-Palet_procesados)>Palet_sin_procesar and Pet_procesado_piezas>Palet_sin_procesar then
286                  Pet_procesado_piezas:=Palet_sin_procesar; (*Si limitan los palets sin procesar*)
287              elsif (24-Palet_procesados)<Palet_sin_procesar and Pet_procesado_piezas>(24-Palet_procesados) then
288                  Pet_procesado_piezas:=(24-Palet_procesados); (*Si limitan los huecos del almacén*)
289              end_if;
290
291          else
292              Pet_procesado_piezas:=0;
293
294          end_if;
295
296      elsif not PROCESANDO then
297
298          (* Si se tienen palets procesados y huecos en la parte de sin procesar *)
299
300          if(36-Palet_sin_procesar)>0 and (palet_procesados)>0 THEN
301
302              (*Estudio de la casuística posible*)
303              if (36-Palet_sin_procesar)>palet_procesados and Pet_desprocesado_piezas>Palet_procesados then
304                  Pet_desprocesado_piezas:=palet_procesados; (*Si limitan los palets procesados*)
305              elsif (36-Palet_sin_procesar)<Palet_sin_procesar and Pet_desprocesado_piezas>(36-Palet_sin_procesar) then
306                  Pet_desprocesado_piezas:=(36-Palet_sin_procesar); (*Si limitan los huecos del almacén*)
307              end_if;
308
309          else
310              Pet_desprocesado_piezas:=0;
311
312          end_if;
313
314      END_IF; (*fin del if de peticiones automáticas *)
315
316      (*Se limita el número máximo de bandejas a 5*)
317      if Nmax_bandeja_instalacion>5 then
318          Nmax_bandeja_instalacion:=5;
319      end_if;
320
321      (* Actualización de variables anteriores (para implementación de flancos en lenguaje ST *)
322      Escritura_peticion_alim_band_ant:=Escritura_peticion_alim_band;
323      mete_pieza_tratada_ant:=mete_pieza_tratada;
324      saca_pieza_sin_tratar_ant:=saca_pieza_sin_tratar;
325      pet_band_alm_ant:=pet_bandejas_leidas[1];
326      pet_band_serv_ant:=pet_bandejas_leidas[2];
327      Almacen_Cinta_libre_ant:=Almacen_Cinta_libre;
328      ent_almacen_ant:=ED_ENT_ALMACEN;
329      sacando_pieza_sin_tratar_ant:=Sacando_pieza_sin_tratar;
330      sacando_pieza_tratada_ant:=sacando_pieza_tratada;

```

Supervisor_errores : [MAST]







Etiquetas truncadas:

Etiqueta	Posición(es)
NbandejasVirtual<>Band_circulacion	(3, 72)
pet_bandejas_LEIDAS[3]=0 AND Band_circulacion<Nmax_bandeja_instalacion	(3, 77)
tramo1[1].tiempo=100	(2, 17)
tramo1[2].tiempo=100	(2, 18)
tramo1[3].tiempo=100	(2, 19)
tramo1[4].tiempo=100	(2, 20)
tramo2[1].tiempo=160	(2, 21)
tramo2[2].tiempo=100	(2, 22)
tramo2[3].tiempo=100	(2, 23)
tramo3[1].tiempo=100	(2, 24)
tramo3[2].tiempo=100	(2, 25)
tramo3[3].tiempo=100	(2, 26)
tramo4[1].tiempo=164	(2, 27)
tramo4[2].tiempo=100	(2, 28)
tramo4[3].tiempo=160	(2, 29)
tramo4[4].tiempo=100	(2, 30)

Asignacion_variables_SCADA : [MAST]

	1	2	3	4	5	6	7	8	9	10	11
1			Asignaciones de los sensores para visualizarse en el SCADA								
2		ED_ENT_ALMACEN							SC_ED_ENT_ALMACEN		
3		ED_ENT_DESVIO_1							SC_ED_ENT_DESVIO_1		
4		ED_ENT_DESVIO_2							SC_ED_ENT_DESVIO_2		
5		ED_ENT_DESVIO_3							SC_ED_ENT_DESVIO_3		
6		ED_ENT_DESVIO_4							SC_ED_ENT_DESVIO_4		
7		ED_ENT_DESVIO_5							SC_ED_ENT_DESVIO_5		
8		ED_ENT_DISP_BANDEJA							SC_ED_ENT_DISP_BANDEJA		
9		ED_ENT_SCARA							SC_ED_ENT_SCARA		
10		ED_PALET_ELEV_1							SC_ED_PALET_ELEV_1		
11		ED_ENT_DESVIO_3							SC_ED_ENT_DESVIO_3		
12		ED_ENT_DESVIO_4							SC_ED_ENT_DESVIO_4		
13		ED_ENT_DESVIO_5							SC_ED_ENT_DESVIO_5		
14		ED_ENT_DISP_BANDEJA							SC_ED_ENT_DISP_BANDEJA		
15		ED_ENT_DESVIO_5							SC_ED_ENT_DESVIO_5		
16		ED_ENT_DISP_BANDEJA							SC_ED_ENT_DISP_BANDEJA		
17		ED_PALET_ELEV_3							SC_ED_PALET_ELEV_3		
18		ED_PALET_ELEV_PUSH_2							SC_ED_PALET_ELEV_PUSH_2		
19		ED_PALET_ELEV_PUSH_4							SC_ED_PALET_ELEV_PUSH_4		
20			Asignaciones de los actuadores								
21											
22		SD_ELEV_DESVIO_1							SC_SD_ELEV_DESVIO_1		
23		SD_ELEV_DESVIO_2							SC_SD_ELEV_DESVIO_2		
24		SD_ELEV_DESVIO_3							SC_SD_ELEV_DESVIO_3		
25		SD_ELEV_DESVIO_4							SC_SD_ELEV_DESVIO_4		
26		SD_MOTOR_1							SC_SD_MOTOR_1		
27		SD_MOTOR_2							SC_SD_MOTOR_2		

reseteos_tramos_tiempos : [MAST]

```
1|      10|      20|      30|      40|      50|      60|      70|      80|      90|     100|     110|
1  if reparacion_fallo_grave or reseteo_tramos then
2
3  Tramo1[1].tiempo:=0;Tramo1[1].bandeja:=0;
4  Tramo1[2].tiempo:=0;Tramo1[2].bandeja:=0;
5  Tramo1[3].tiempo:=0;Tramo1[3].bandeja:=0;
6  Tramo1[4].tiempo:=0;Tramo1[4].bandeja:=0;
7  Tramo2[1].tiempo:=0;Tramo2[1].bandeja:=0;
8  Tramo2[2].tiempo:=0;Tramo2[2].bandeja:=0;
9  Tramo2[3].tiempo:=0;Tramo2[3].bandeja:=0;
10 Tramo3[1].tiempo:=0;Tramo3[1].bandeja:=0;
11 Tramo3[2].tiempo:=0;Tramo3[2].bandeja:=0;
12 Tramo3[3].tiempo:=0;Tramo3[3].bandeja:=0;
13 Tramo4[1].tiempo:=0;Tramo4[1].bandeja:=0;
14 Tramo4[2].tiempo:=0;Tramo4[2].bandeja:=0;
15 Tramo4[3].tiempo:=0;Tramo4[3].bandeja:=0;
16 Tramo4[4].tiempo:=0;Tramo4[4].bandeja:=0;
17
18 end_if;
```

Asignacion_variables_SCADA : [MAST]

	1	2	3	4	5	6	7	8	9	10	11
1			Asignaciones de los sensores para visualizarse en el SCADA								
2		ED_ENT_ALMACEN							SC_ED_ENT_ALMACEN		
3		ED_ENT_DESVIO_1							SC_ED_ENT_DESVIO_1		
4		ED_ENT_DESVIO_2							SC_ED_ENT_DESVIO_2		
5		ED_ENT_DESVIO_3							SC_ED_ENT_DESVIO_3		
6		ED_ENT_DESVIO_4							SC_ED_ENT_DESVIO_4		
7		ED_ENT_DESVIO_5							SC_ED_ENT_DESVIO_5		
8		ED_ENT_DISP_BANDEJA							SC_ED_ENT_DISP_BANDEJA		
9		ED_ENT_SCARA							SC_ED_ENT_SCARA		
10		ED_PALET_ELEV_1							SC_ED_PALET_ELEV_1		
11		ED_ENT_DESVIO_3							SC_ED_ENT_DESVIO_3		
12		ED_ENT_DESVIO_4							SC_ED_ENT_DESVIO_4		
13		ED_ENT_DESVIO_5							SC_ED_ENT_DESVIO_5		
14		ED_ENT_DISP_BANDEJA							SC_ED_ENT_DISP_BANDEJA		
15		ED_ENT_DESVIO_5							SC_ED_ENT_DESVIO_5		
16		ED_ENT_DISP_BANDEJA							SC_ED_ENT_DISP_BANDEJA		
17		ED_PALET_ELEV_3							SC_ED_PALET_ELEV_3		
18		ED_PALET_ELEV_PUSH_2							SC_ED_PALET_ELEV_PUSH_2		
19		ED_PALET_ELEV_PUSH_4							SC_ED_PALET_ELEV_PUSH_4		
20			Asignaciones de los actuadores								
21											
22		SD_ELEV_DESVIO_1							SC_SD_ELEV_DESVIO_1		
23		SD_ELEV_DESVIO_2							SC_SD_ELEV_DESVIO_2		
24		SD_ELEV_DESVIO_3							SC_SD_ELEV_DESVIO_3		
25		SD_ELEV_DESVIO_4							SC_SD_ELEV_DESVIO_4		
26		SD_MOTOR_1							SC_SD_MOTOR_1		
27		SD_MOTOR_2							SC_SD_MOTOR_2		

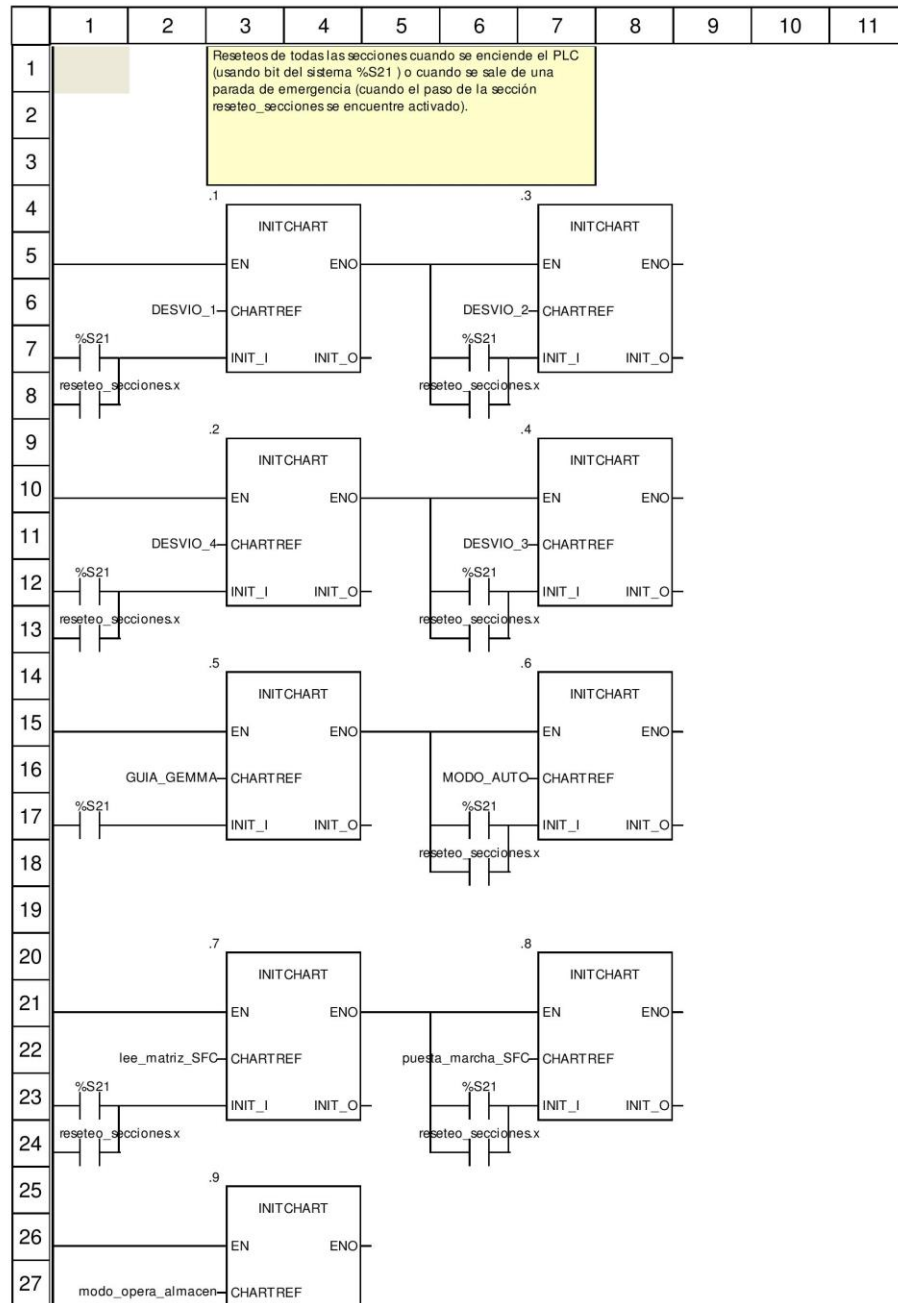
	1	2	3	4	5	6	7	8	9	10	11
59	Tramo1[1].bandeja							SC_Bandeja11			
60	Tramo1[2].bandeja							SC_Bandeja12			
61	Tramo1[3].bandeja							SC_Bandeja13			
62	Tramo1[4].bandeja							SC_Bandeja14			
63	Tramo2[1].bandeja							SC_Bandeja21			
64	Tramo2[2].bandeja							SC_Bandeja22			
65	Tramo2[3].bandeja							SC_Bandeja23			
66	Tramo3[1].bandeja							SC_Bandeja31			
67	Tramo3[2].bandeja							SC_Bandeja32			
68	Tramo3[3].bandeja							SC_Bandeja33			
69	Tramo4[1].bandeja							SC_Bandeja41			
70	Tramo4[2].bandeja							SC_Bandeja42			
71	Tramo4[3].bandeja							SC_Bandeja43			
72	Tramo4[4].bandeja							SC_Bandeja44			
73											
74											
75											
76	Tramo1[1].palet							SC_Palet11			
77	Tramo1[2].palet							SC_Palet12			
78	Tramo1[3].palet							SC_Palet13			
79	Tramo1[4].palet							SC_Palet14			
80	Tramo2[1].palet							SC_Palet21			
81	Tramo2[2].palet							SC_Palet22			
82	Tramo2[3].palet							SC_Palet23			
83	Tramo3[1].palet							SC_Palet31			
84	Tramo3[2].palet							SC_Palet32			
85	Tramo3[3].palet							SC_Palet33			
86	Tramo4[1].palet							SC_Palet41			
87	Tramo4[2].palet							SC_Palet42			
88	Tramo4[3].palet							SC_Palet43			
89	Tramo4[4].palet							SC_Palet44			

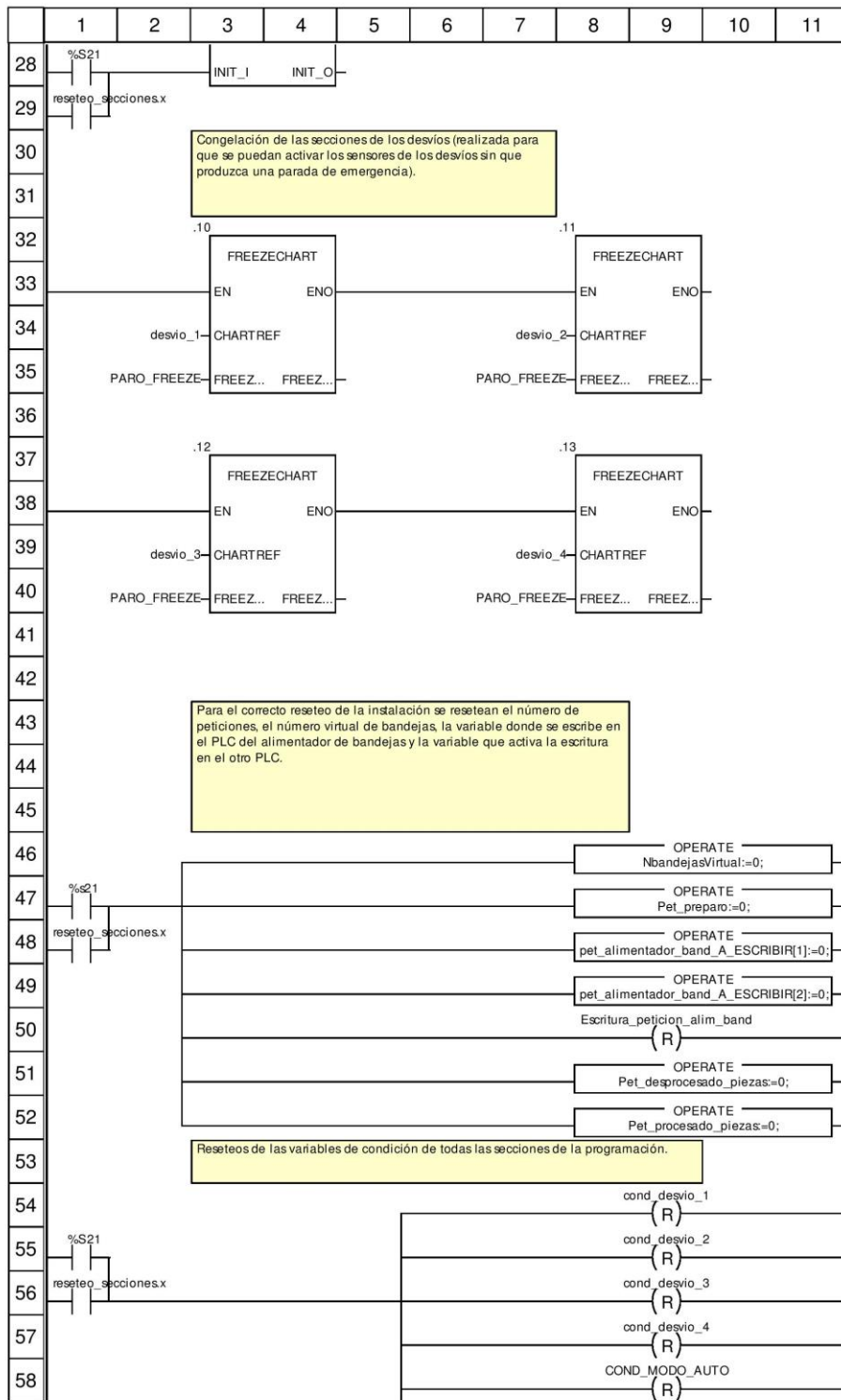
	1	2	3	4	5	6	7	8	9	10	11
90											
91											
92											
93	Tramo1[1].procesado							SC_Procesado11			
94	Tramo1[2].procesado							SC_Procesado12			
95	Tramo1[3].procesado							SC_Procesado13			
96	Tramo1[4].procesado							SC_Procesado14			
97	Tramo2[1].procesado							SC_Procesado21			
98	Tramo2[2].procesado							SC_Procesado22			
99	Tramo2[3].procesado							SC_Procesado23			
100	Tramo3[1].procesado							SC_Procesado31			
101	Tramo3[2].procesado							SC_Procesado32			
102	Tramo3[3].procesado							SC_Procesado33			
103	Tramo4[1].procesado							SC_Procesado41			
104	Tramo4[2].procesado							SC_Procesado42			
105	Tramo4[3].procesado							SC_Procesado43			
106	Tramo4[4].procesado							SC_Procesado44			

Etiquetas truncadas:

Etiqueta	Posición(es)
SC_SD_PILOTO_MODO_BUSQUEDA	(9, 31)
SC_SD_PILOTO_PARADA_MOTORES	(9, 32)

Reseteos_congelacion_secciones : [MAST]

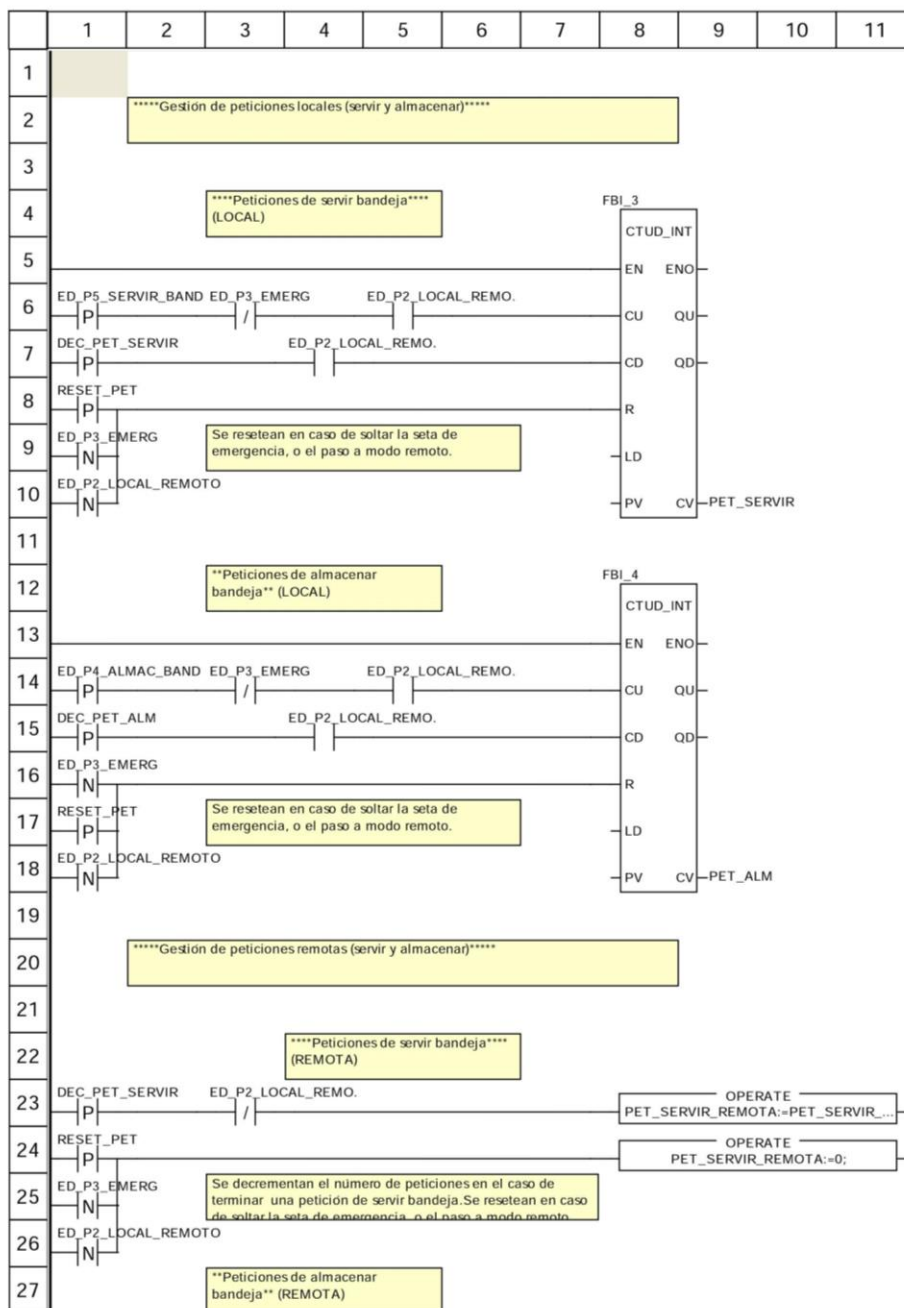


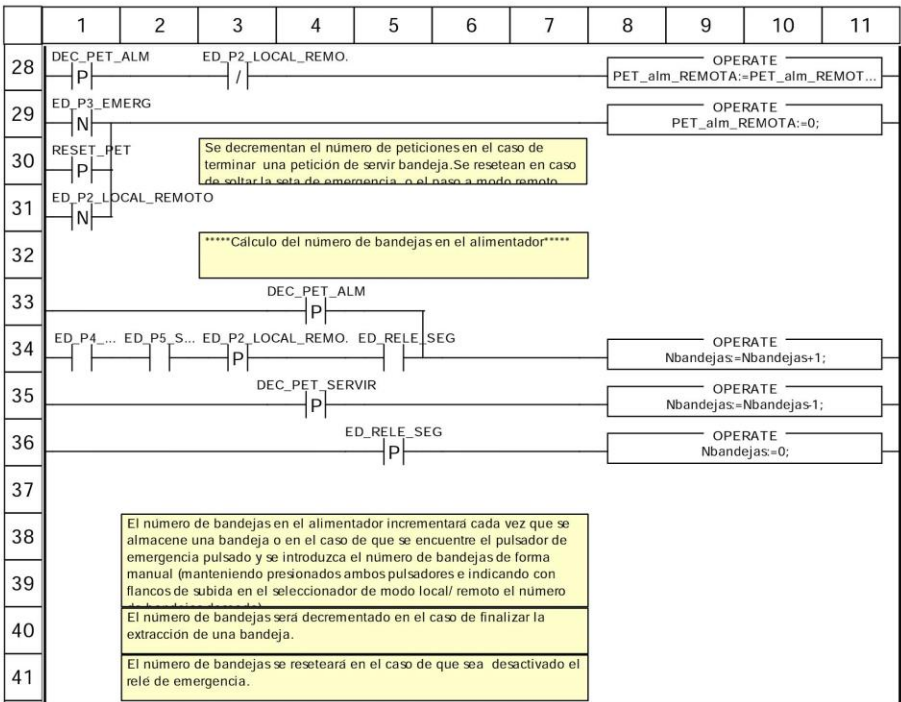


	1	2	3	4	5	6	7	8	9	10	11
59						COND_PUESTA_MARCHA (R)					
60						COND_PROG_RET_AUTO (R)					
61						cond_pet_bandejas (R)					
62						parq_freeze (R)					

ANEXO B: PROGRAMACIÓN DEL ALIMENTADOR DE BANDEJAS

Gestor_peticiones : [MAST]





Etiquetas truncadas:

Etiqueta	Posición(es)
ED_P2_LOCAL_REMOTO	(5, 6) (4, 7) (5, 14) (4, 15) (3, 23) (3, 28) (3, 34)
ED_P4_ALMAC_BAND	(1, 34)
ED_P5_SERVIR_BAND	(2, 34)
PET_SERVIR_REMOTA:=PET_SERVIR_REMOTA-1;	(8, 23)
PET_alm_REMOTA:=PET_alm_REMOTA-1;	(8, 28)

Chart : [MAST - Guia_GEMMA]

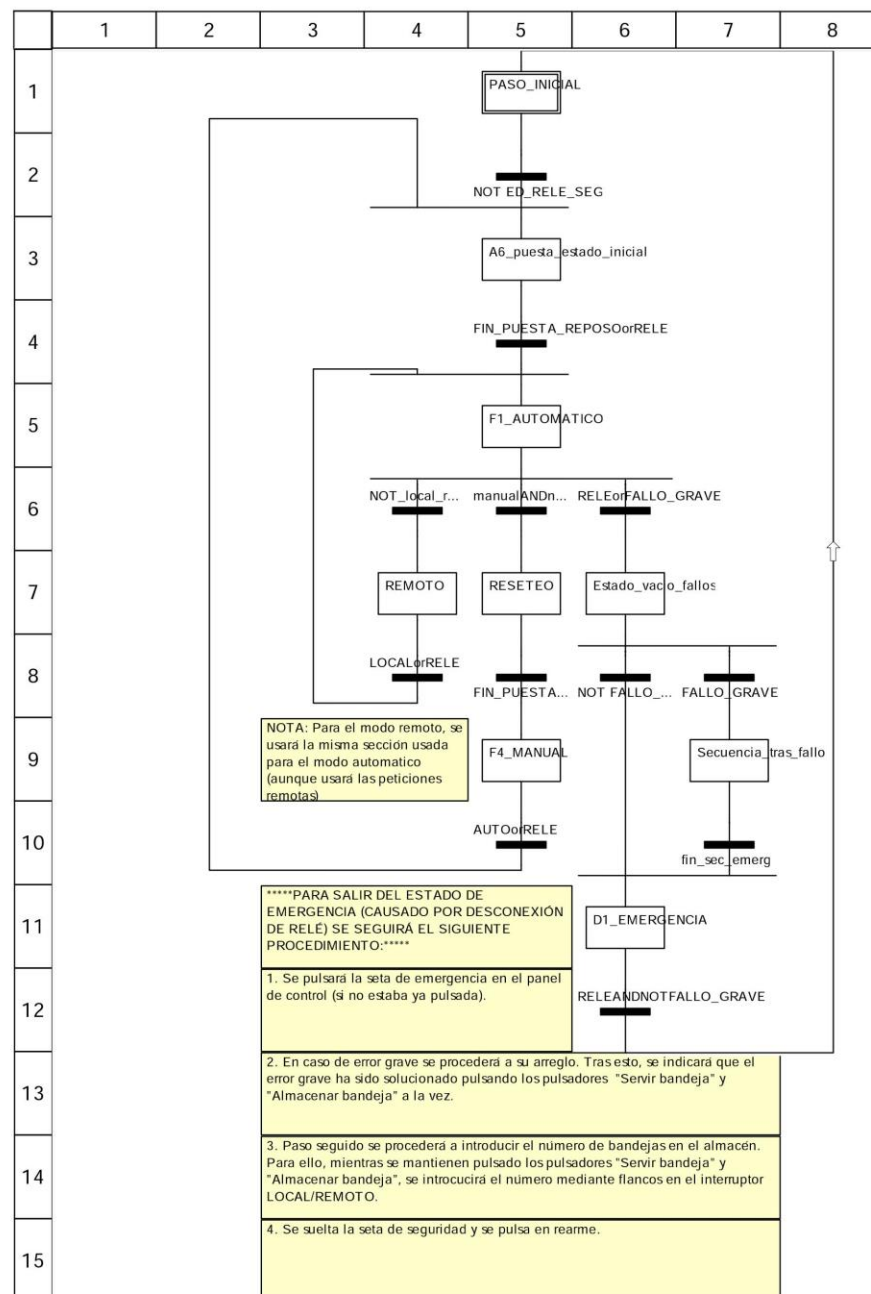
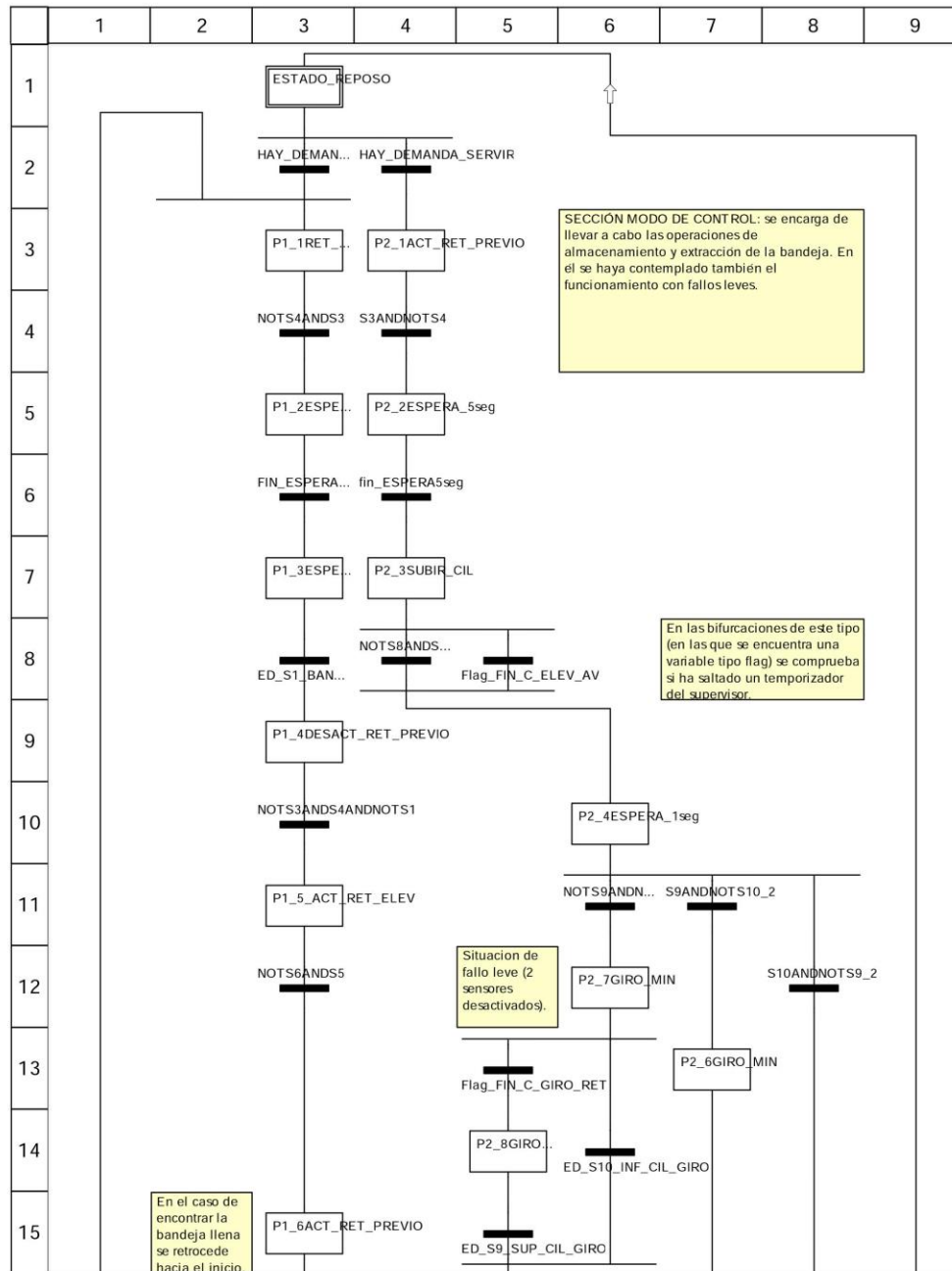
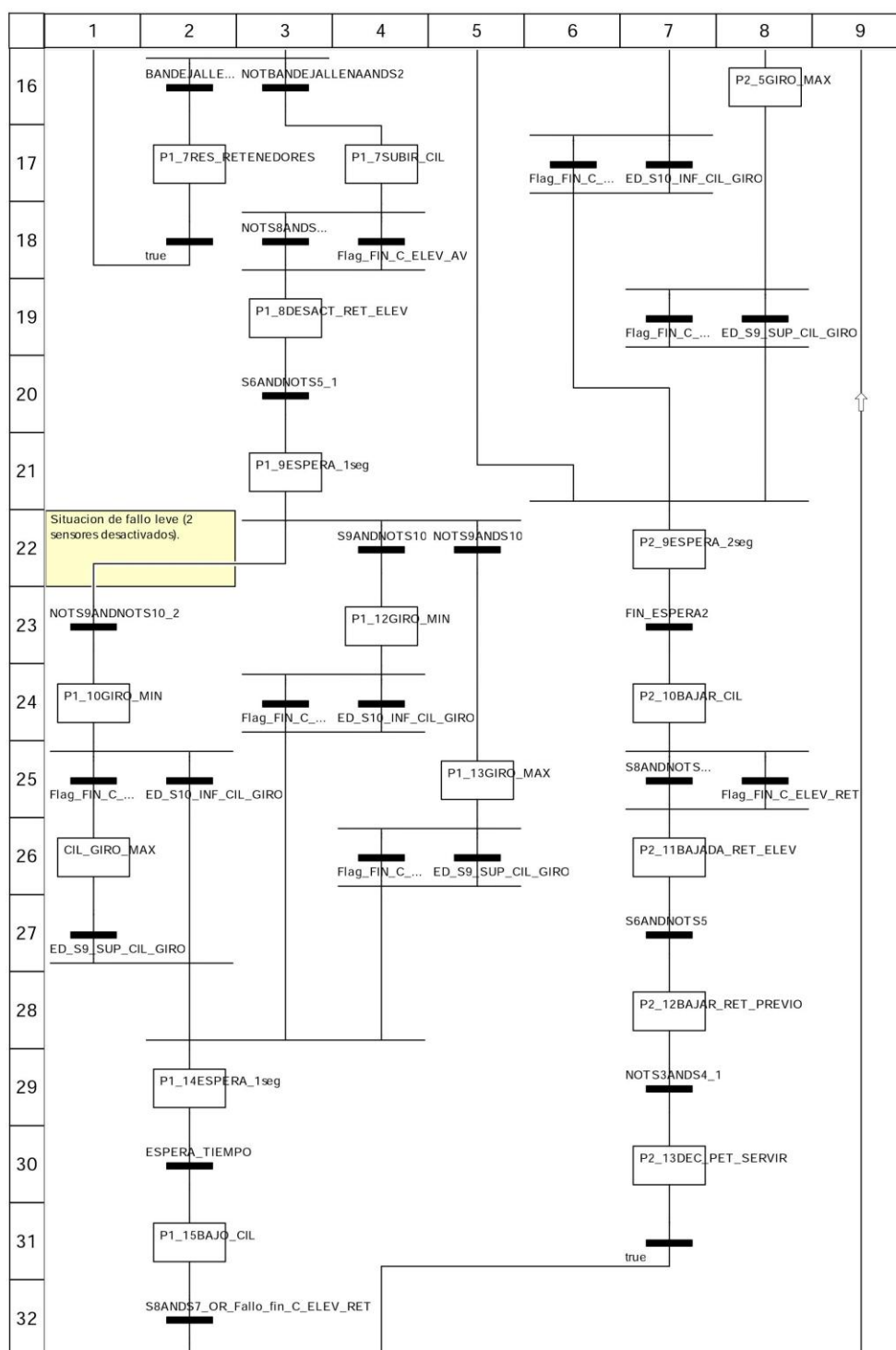


Chart : [MAST - Modo_control]





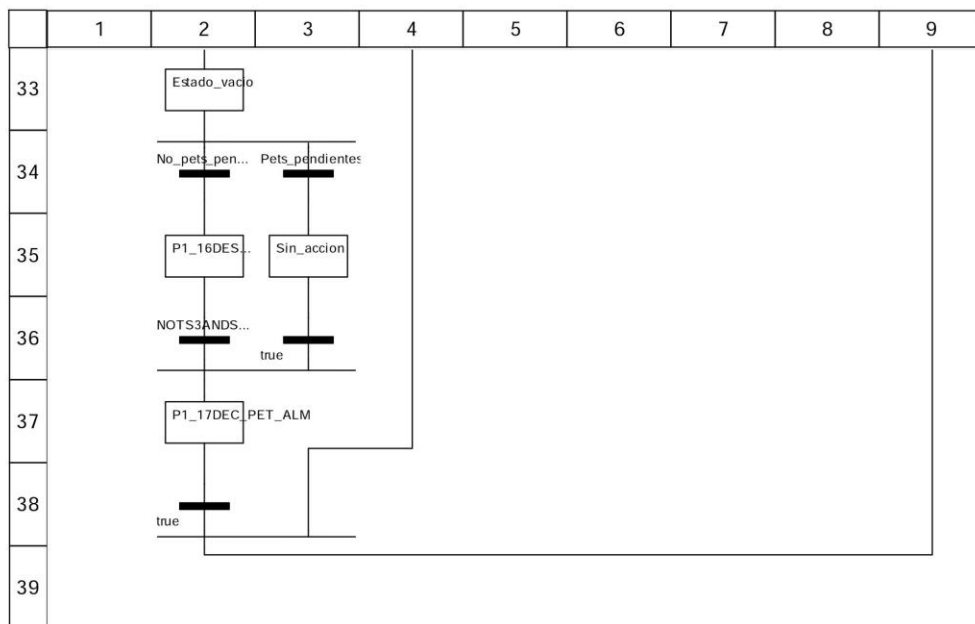
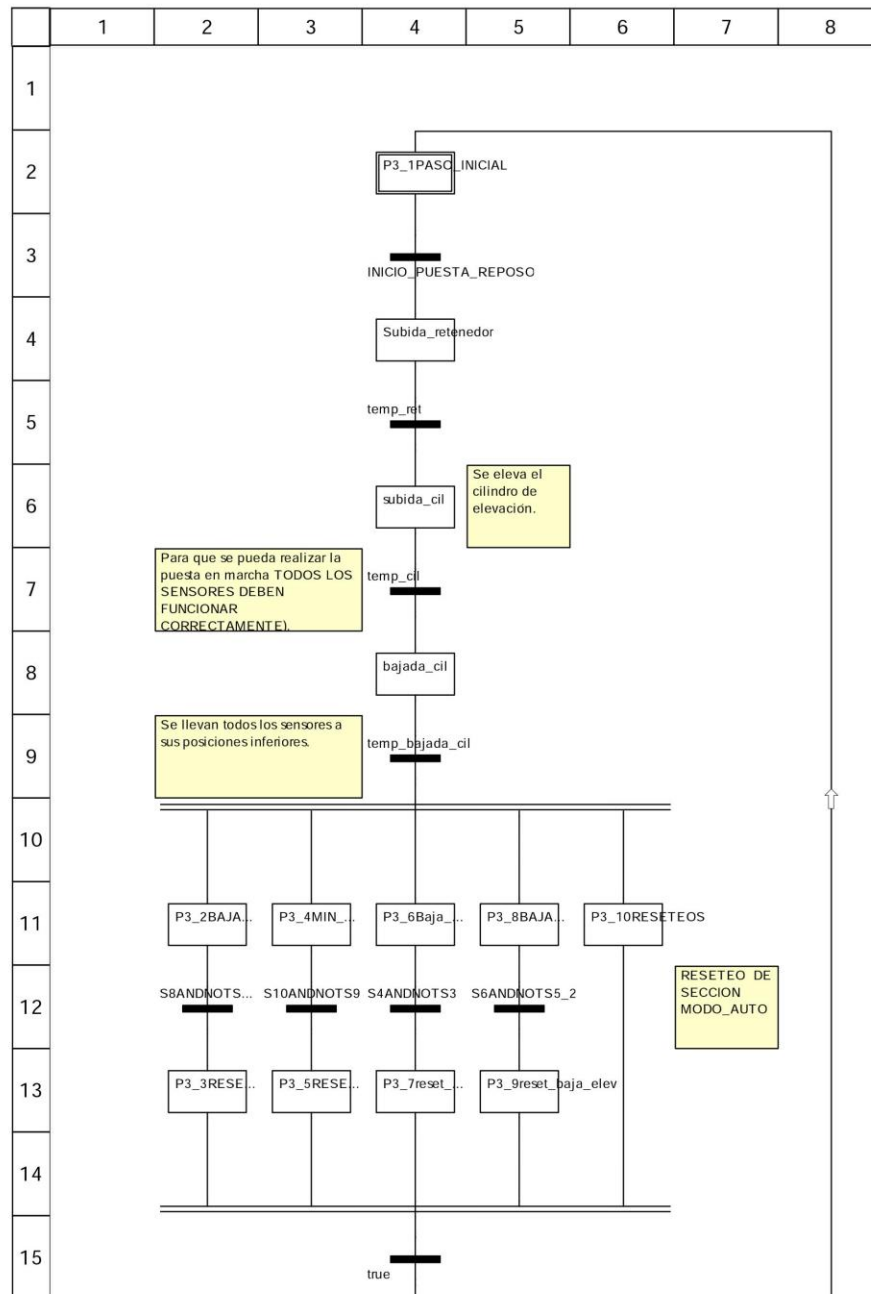
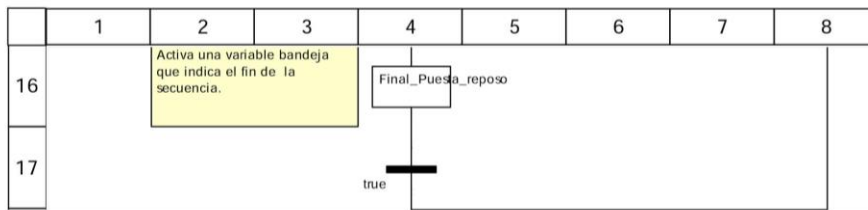
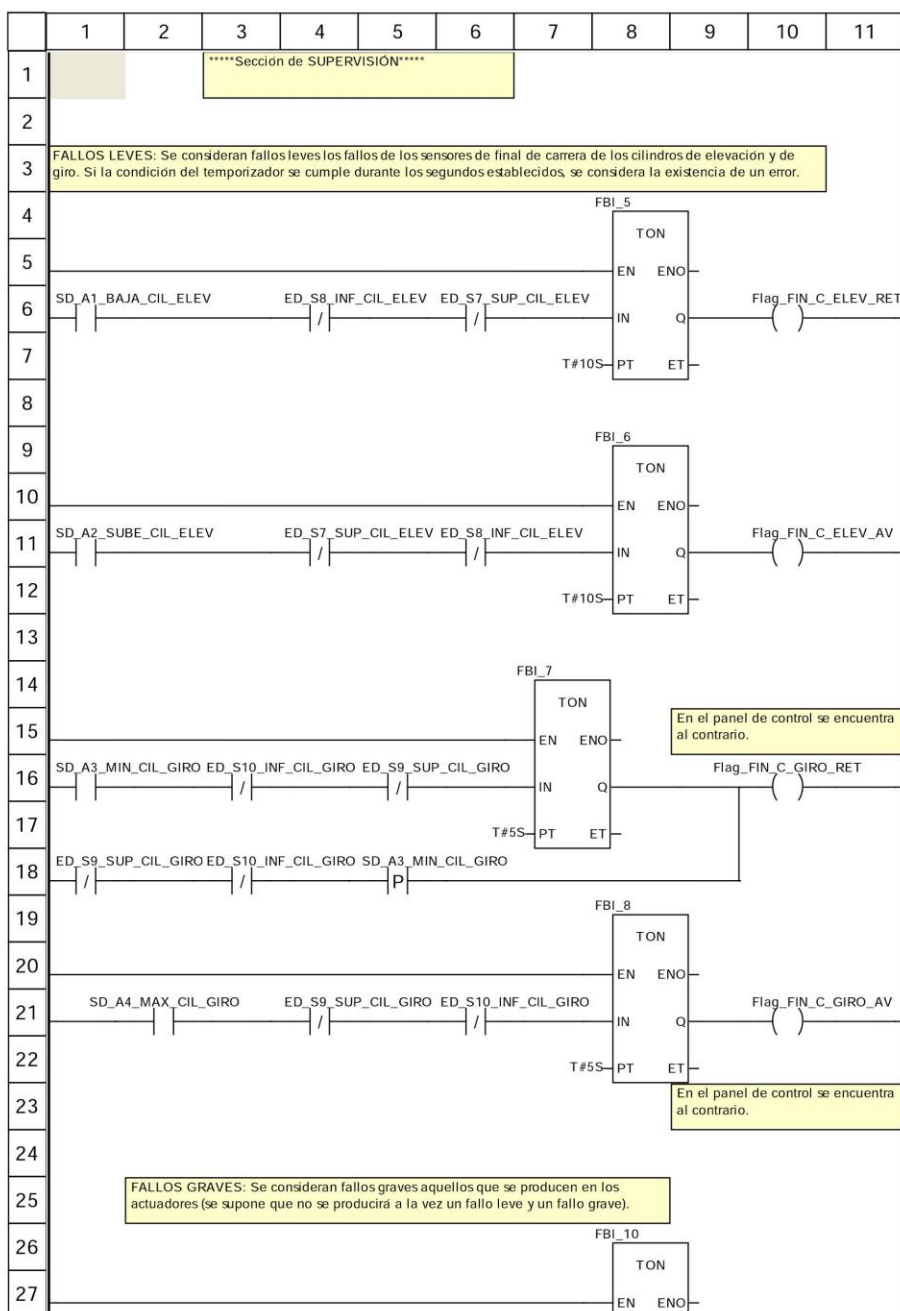


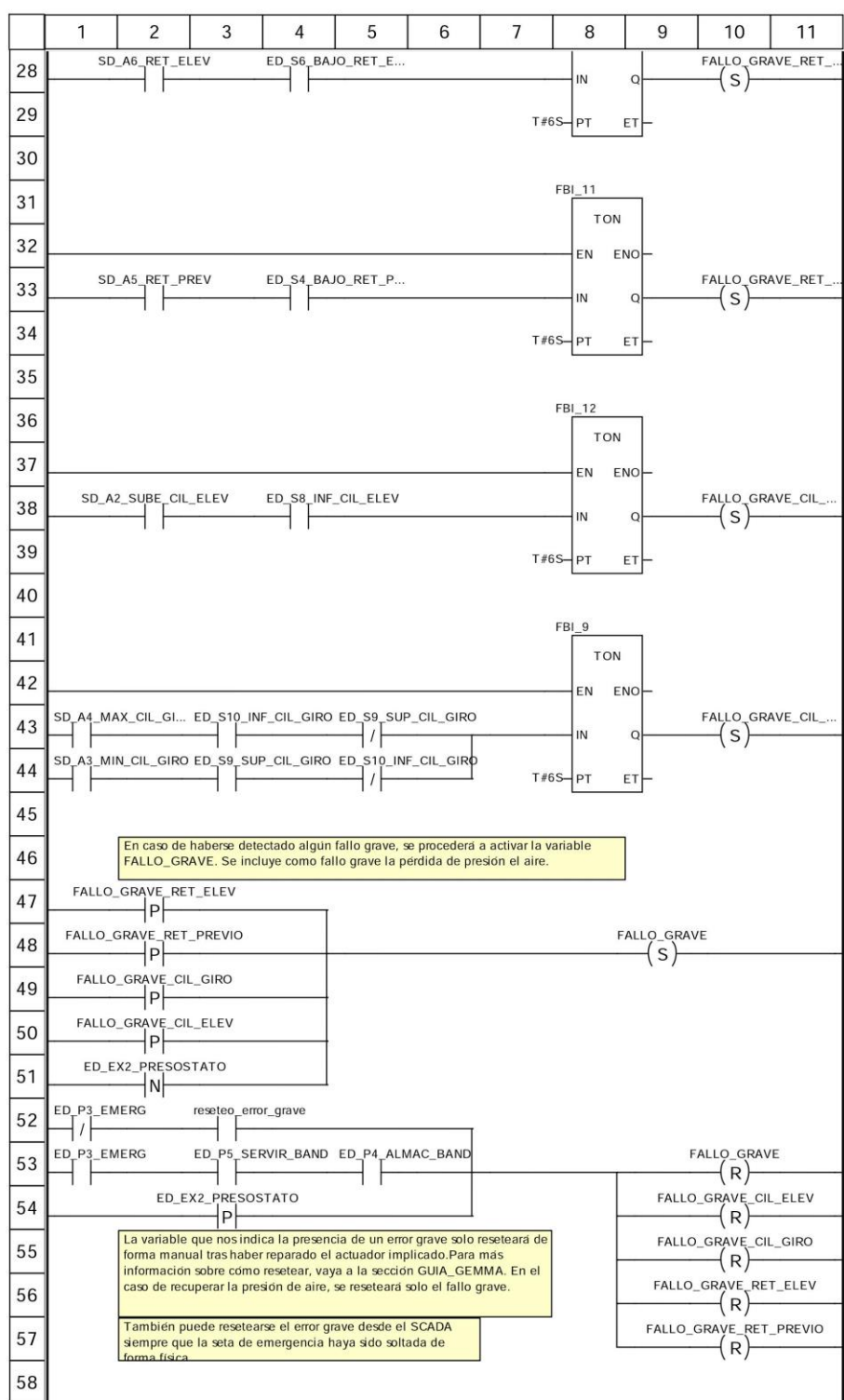
Chart : [MAST - Puesta_en_reposo]

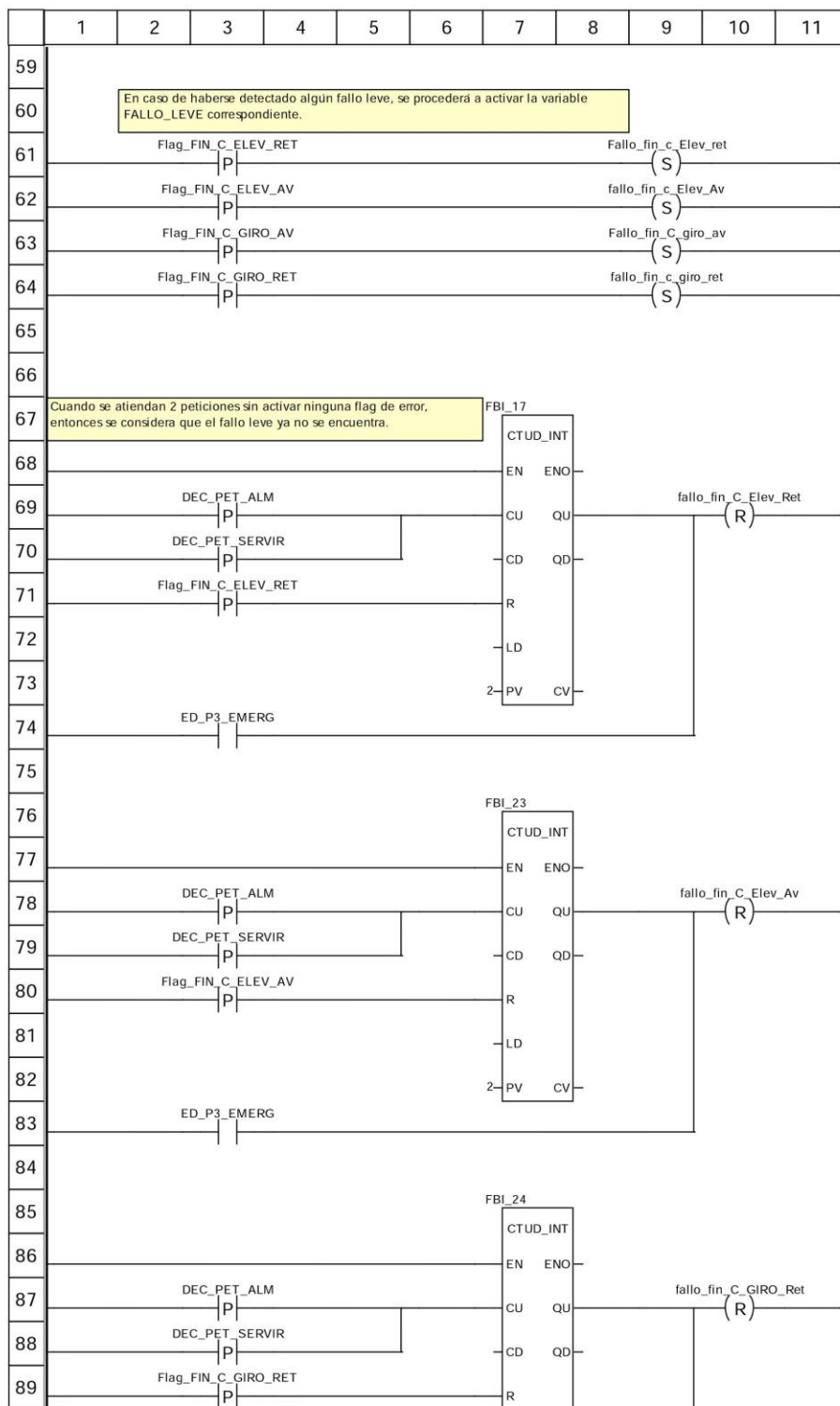


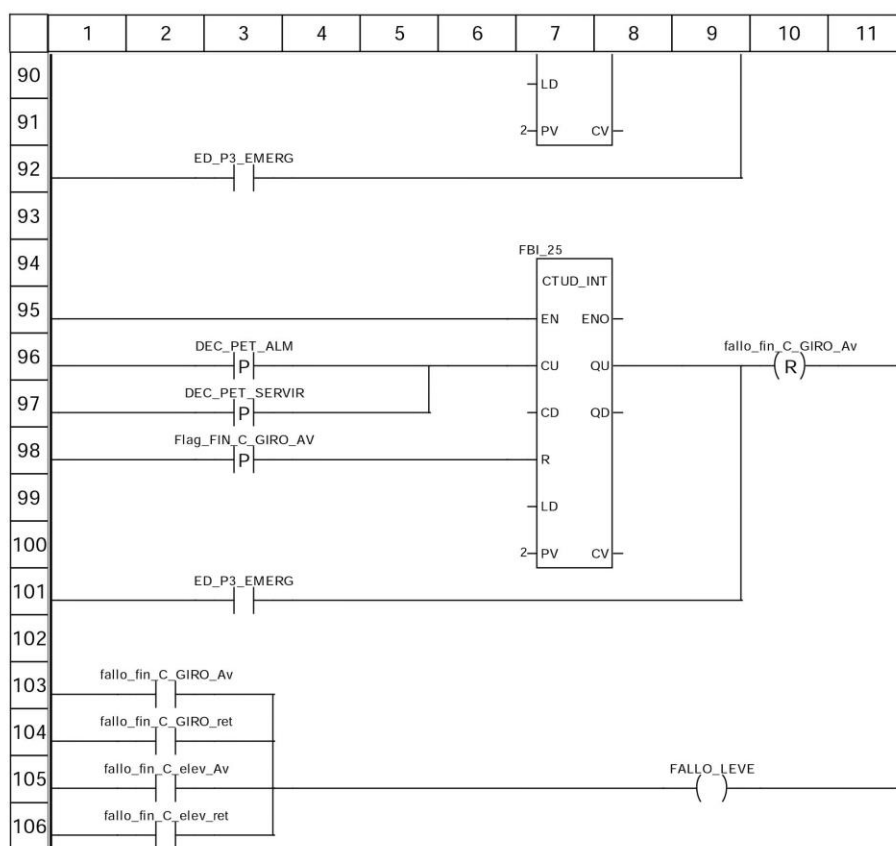


Supervisor : [MAST]







**Etiquetas truncadas:**

Etiqueta	Posición(es)
ED_S4_BAJO_RET_PREV	(4, 33)
ED_S6_BAJO_RET_ELEV	(4, 28)
FALLO_GRAVE_CIL_ELEV	(10, 38)
FALLO_GRAVE_CIL_GIRO	(10, 43)
FALLO_GRAVE_RET_ELEV	(10, 28)
FALLO_GRAVE_RET_PREVIO	(10, 33)
SD_A4_MAX_CIL_GIRO	(1, 43)

Reseteos_secciones : [MAST]

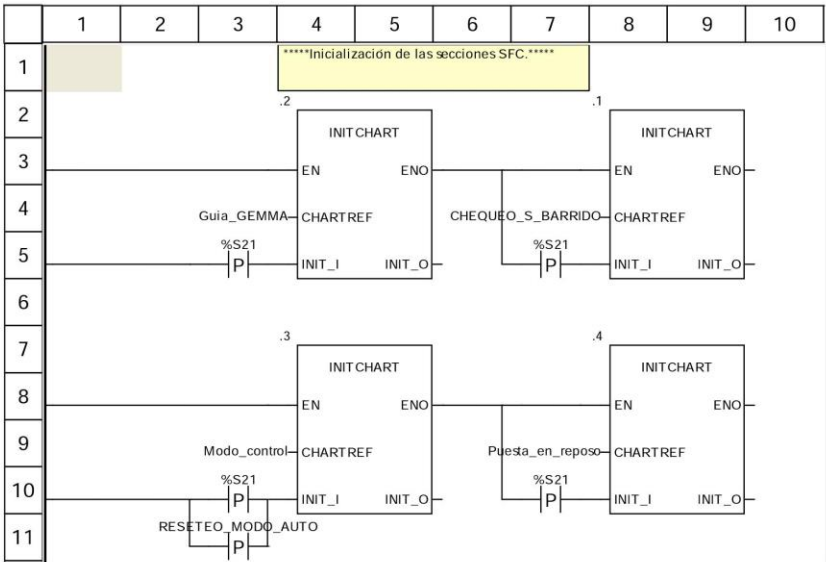


Chart : [MAST - CHEQUEO_S_BARRIDO]

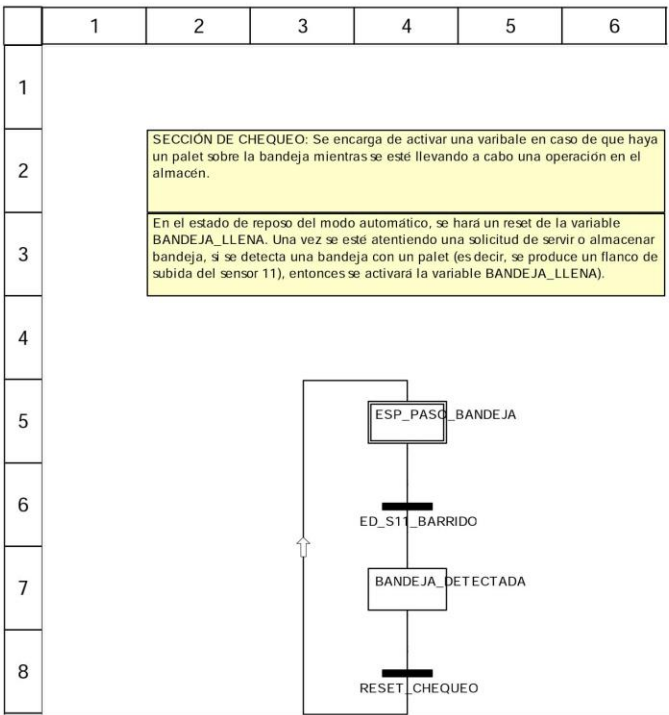
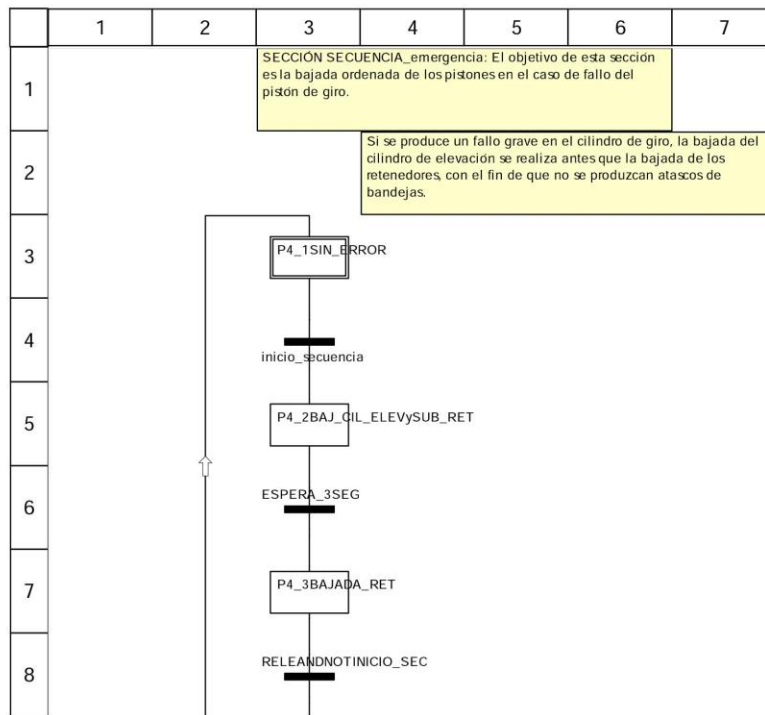
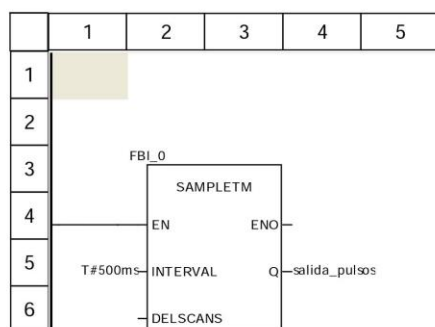


Chart : [MAST - SECUENCIA_emergencia]



Generador_pulsos_visual : [MAST]



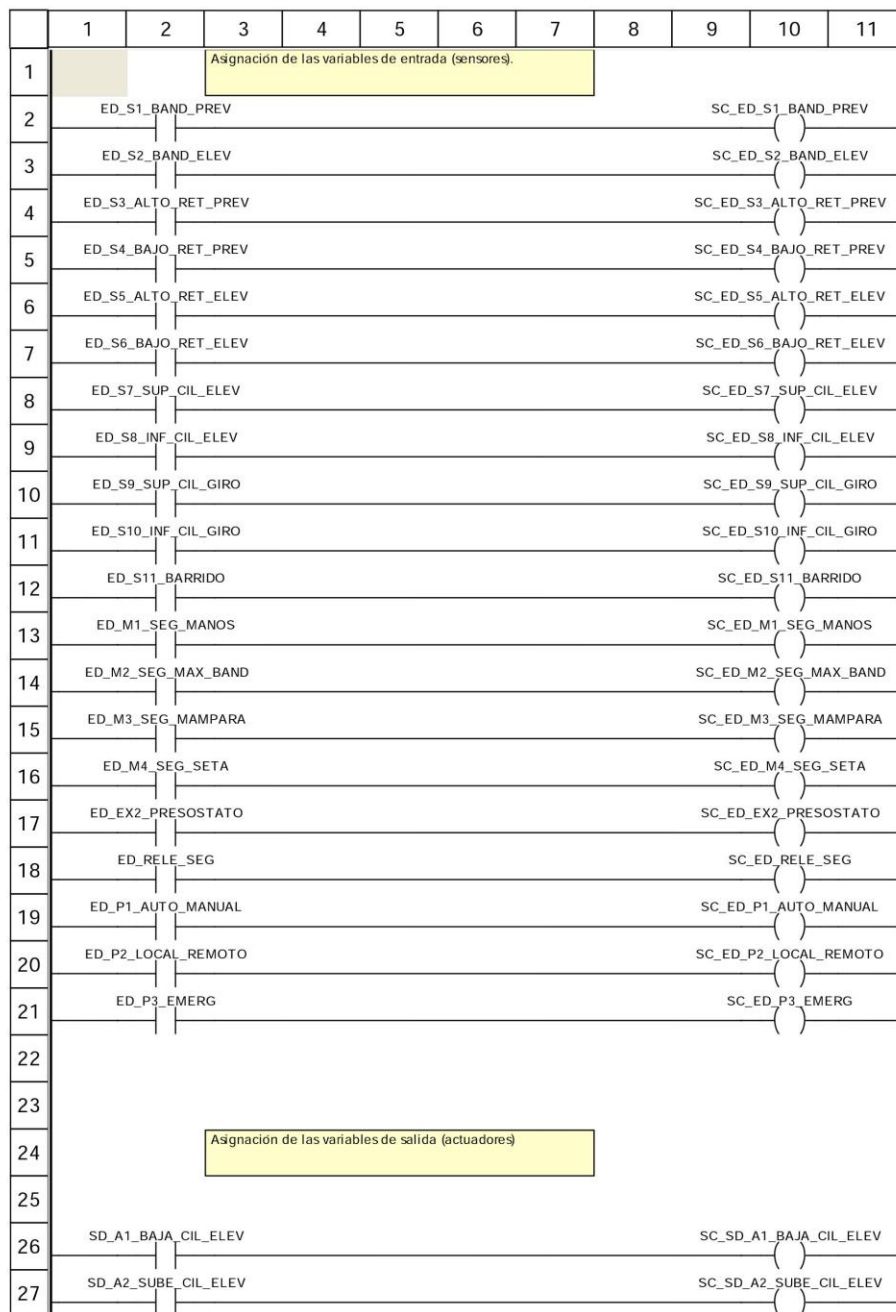
visualizacion_SCADA : [MAST]

```

1|      10|      20|      30|      40|      50|      60|      70|      80|      90|     100|     110|
1  sensor_barrido:=ED_S11_BARRIDO;
2
3
4
5  (* MOVIMIENTO DE BANDEJA DESDE EL RETENEDOR PREVIO AL DE ELEVACION *)
6  IF (ED_S1_BAND_PREV AND ED_S4_BAJO_RET_PREV) THEN TRAMO_PREV_A_ELEV:=1;TIME_PREV_A_ELEV:=0;
7
8  END_IF;
9
10 IF ED_S2_BAND_ELEV THEN TRAMO_PREV_A_ELEV:=0;
11 END_IF;
12
13
14
15
16 IF TRAMO_PREV_A_ELEV THEN
17     IF salida_pulsos THEN TIME_PREV_A_ELEV:=TIME_PREV_A_ELEV+1;
18     END_IF;
19 END_IF;
20
21
22 (* MOVIMIENTO DEL CILINDRO DESDE LA POSICIÓN INICIAL A LA FINAL *)
23 IF (ED_S8_INF_CIL_ELEV AND NOT ED_S7_SUP_CIL_ELEV) THEN TRAMO_CIL_INF_A_SUP:=1;TIME_INF_A_SUP:=0;
24
25 END_IF;
26
27 IF (NOT ED_S8_INF_CIL_ELEV AND ED_S7_SUP_CIL_ELEV) THEN TRAMO_CIL_SUP_A_INF:=1;TIME_SUP_A_INF:=0;
28
29 END_IF;
30
31
32 IF TRAMO_CIL_INF_A_SUP THEN
33     IF salida_pulsos THEN TIME_INF_A_SUP:=TIME_INF_A_SUP+1;
34     END_IF;
35 END_IF;
36
37 IF TRAMO_CIL_SUP_A_INF THEN
38     IF salida_pulsos THEN TIME_SUP_A_INF:=TIME_SUP_A_INF+1;
39     END_IF;
40 END_IF;
41
42

```


SCADA_Asignacion_variables : [MAST]



	1	2	3	4	5	6	7	8	9	10	11
28	SD_A3_MIN_CIL_GIRO								SC_SD_A3_MIN_CIL_GIRO ()		
29	SD_A4_MAX_CIL_GIRO								SC_SD_A4_MAX_CIL_GIRO ()		
30	SD_A5_RET_PREV								SC_SD_A5_RET_PREV ()		
31	SD_A6_RET_ELEV								SC_SD_A6_RET_ELEV ()		

ANEXO C: PROGRAMACIÓN DEL SISTEMA ROBOTIZADO DE ALMACENAJE

```
PROGRAM INIC
*****
SET    PUMAR = 0
SET    SITUA[1] = 0
SET    SITUA[2] = 1
SET    SITUA[3] = 0
SET    SITUA[4] = 0
SET    SITUA[5] = 0
SET    SITUA[6] = 1
SET    SENT = 0
SET    OUT[4] = 1
SPEED  10
MOVED  CIM[499]
SET    CNS=LEVL5 * 2
FOR    J = 1 TO CNS
  SET   AUX=CELLS * J
  IF    SENT = 0
    SET  CNST=AUX + 400
    SET  CNST=CNST - 5
  ELSE
    SET  CNST=400 + AUX
  ENDIF
  IF    CNST = 437
    MOVED CIM[CNST]
  ENDIF
```

```
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SET    CNST=CNST - 100
FOR    I = 1 TO CELLS
  MOVED  CIM[CNST]
  SET    INDI=CNST - 200
  IF     IN[1] = 1
    SET    LLENO[INDI] = 0
  ELSE
    SET    LLENO[INDI] = 1
  ENDIF
  PRINTLN
  PRINT  LLENO[INDI]
  IF     SENT = 0
    SET    CNST=CNST + 1
  ELSE
    SET    CNST=CNST - 1
  ENDIF
ENDFOR
PRINTLN
IF     SENT = 0
  SET    CNST=CNST - 1
ELSE
  SET    CNST=CNST + 1
ENDIF
SET    CNST=CNST + 100
MOVED  CIM[CNST]
SET    CNST=CNST + 100
IF     SENT = 1
  SET    SENT = 0
ELSE
  SET    SENT = 1
ENDIF
ENDFOR
MOVED  CIM[499]
SET    OUT[4] = 0
SET    SITUA[1] = 1
```

```
SET    SITUA[2] = 0
SET    SITUA[3] = 0
SET    SITUA[4] = 0
SET    SITUA[5] = 0
SET    SITUA[6] = 0
SET    PUMAR = 1
END

      PROGRAM ESTAD
      *****

SET    INDEX = 36
PRINTLN
PRINTLN "Piezas no procesadas (del 1 al 36)"
PRINTLN
FOR    J = 1 TO LEVLS
  FOR    I = 1 TO CELLS
    PRINT LLENO[INDEX]
    SET    INDEX=INDEX - 1
  ENDFOR
  PRINTLN
ENDFOR
PRINTLN
PRINTLN
PRINTLN "Piezas procesadas (del 49 al 72)"
PRINTLN
SET    INDEX = 72
FOR    I = 1 TO 4
  FOR    J = 1 TO CELLS
    PRINT LLENO[INDEX]
    SET    INDEX=INDEX - 1
  ENDFOR
  PRINTLN
ENDFOR
PRINTLN
PRINTLN "Piezas defectuosas (del 37al 49)"
PRINTLN
FOR    I = 1 TO 2
  FOR    J = 1 TO CELLS
```

```
PRINT  LLENO[INDEX]
SET    INDEX=INDEX - 1
ENDFOR
PRINTLN
ENDFOR
END

      PROGRAM  METE
      *****

SET    SITUA[1] = 0
SET    SITUA[2] = 0
SET    SITUA[3] = 1
SET    SITUA[4] = 0
SET    SITUA[5] = 0
SET    SITUA[6] = 0
SPEED  15
OPEN
PRINT  "Solicitud de meter b. tratada"
SET    LIBRE = 0
SET    INDEX = 49
FOR    J = 1 TO 4
  FOR    I = 1 TO CELLS
    IF    LLENO[INDEX] = 0
      SET    LIBRE = INDEX
    ENDIF
    SET    INDEX=INDEX + 1
  ENDFOR
ENDFOR
ENDFOR
IF    LIBRE <> 0
  SET    OUT[4] = 1
  MOVED  CINT3
  MOVED  CINT2
  SPEED  5
  MOVED  CINT1
  CLOSE
  MOVED  CINT2
  SET    SITUA[1] = 0
```

```
SET    SITUA[2] = 0
SET    SITUA[3] = 1
SET    SITUA[4] = 1
SET    SITUA[5] = 1
SET    SITUA[6] = 1
SPEED  15
MOVED  CINT3
SET    CNST=LIBRE + 400
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
OPEN
SET    CNST=CNST + 100
SET    LLEN0[LIBRE] = 1
MOVED  CIM[CNST]
SET    CNST=CNST + 100
MOVED  CIM[CNST]
SET    CNST=CNST + 100
MOVED  CIM[CNST]
MOVED  CIM[499]
SET    OUT[4] = 0
SET    SITUA[1] = 1
SET    SITUA[2] = 0
SET    SITUA[3] = 0
SET    SITUA[4] = 0
SET    SITUA[5] = 0
SET    SITUA[6] = 0
ELSE
SET    SITUA[1] = 0
SET    SITUA[2] = 1
SET    SITUA[3] = 1
SET    SITUA[4] = 1
SET    SITUA[5] = 0
SET    SITUA[6] = 0
```

```
DELAY 400
ENDIF
END

PROGRAM EXTRA
*****

SET SITUA[1] = 0
SET SITUA[2] = 0
SET SITUA[3] = 0
SET SITUA[4] = 0
SET SITUA[5] = 0
SET SITUA[6] = 0
PRINTLN
PRINT "Solicitud de extraccion b. tratada"
SPEED 15
OPEN
SET BAND = 0
SET INDEX = 49
FOR I = 1 TO 4
  FOR J = 1 TO CELLS
    IF LLENO[INDEX] = 1
      SET BAND = INDEX
    ENDIF
    SET INDEX=INDEX + 1
  ENDFOR
ENDFOR
IF BAND <> 0
  PRINTLN
  PRINT "Encontrada bandeja en " BAND
  SET CNST=400 + BAND
  SET OUT[4] = 1
  MOVED CIM[CNST]
  SET CNST=CNST - 100
  MOVED CIM[CNST]
  SET CNST=CNST - 100
  MOVED CIM[CNST]
  SPEED 5
```



```
SET    CNST=CNST - 100
MOVED  CIM[CNST]
CLOSE
SET    CNST=CNST - 100
SET    LLENO[CNST] = 0
SET    CNST=CNST + 200
MOVED  CIM[CNST]
SPEED  15
SET    CNST=CNST + 100
MOVED  CIM[CNST]
SET    CNST=CNST + 100
MOVED  CIM[CNST]
MOVED  CINT3
MOVED  CINT2
SPEED  5
MOVED  CINT1
OPEN
MOVED  CINT2
SET    SITUA[1] = 0
SET    SITUA[2] = 0
SET    SITUA[3] = 1
SET    SITUA[4] = 1
SET    SITUA[5] = 1
SET    SITUA[6] = 1
SPEED  15
MOVED  CINT3
MOVED  CIM[499]
PRINTLN "Proceso terminado"
SET    OUT[4] = 0
SET    SITUA[1] = 1
SET    SITUA[2] = 0
SET    SITUA[3] = 0
SET    SITUA[4] = 0
SET    SITUA[5] = 0
SET    SITUA[6] = 0
ELSE
SET    SITUA[1] = 0
SET    SITUA[2] = 1
```

```
SET    SITUA[3] = 1
SET    SITUA[4] = 1
SET    SITUA[5] = 0
SET    SITUA[6] = 0
PRINTLN "ERROR: no hay piezas"
DELAY  400
ENDIF
END

      PROGRAM RECEI
      *****

SET    FIN = 0
SET    ERREC = 0
PRINT  "Recepcion de mensaje: "
SET    I = 1
LABEL  2
GETCOM 0,VAR
IF     I < 12
  IF    VAR = 48
    SET  MENSA[I] = 0
    PRINT MENSA[I]
    SET  I=I + 1
    GOTO 2
  ELSE
    IF    VAR = 49
      SET  MENSA[I] = 1
      PRINT MENSA[I]
      SET  I=I + 1
      GOTO 2
    ELSE
      IF    VAR = 47
        PRINT "Recepcion completa"
      ELSE
        PRINTLN " Error: caracter no conocido"
        PRINTLN "valor de caracter: " VAR
        SET  ERREC = 1
      ENDIF
    ENDIF
  ENDIF
ENDIF
```

```
ENDIF
ENDIF
ELSE
  PRINT  " Error:excedido longitud maxima"
  SET   ERREC = 1
ENDIF
PRINTLN
LABEL  3
SET   MENSA[I] = 0
SET   I=I + 1
IF    I < 14
  GOTO 3
ENDIF
SET   FIN = 1
END
```

```
      PROGRAM CRC
      *****

SET   FIN = 0
SET   CONT = 0
SET   CORRE = 0
PRINTLN
PRINT  " Mensaje: "
FOR    I = 1 TO 14
  PRINT  MENSA[I]
ENDFOR
FOR    I = 1 TO 4
  SET   DESP[I] = 0
ENDFOR
LABEL  1
SET   J=CONT + 1
SET   EREG = MENSA[J]
SET   SREG = DESP[1]
FOR    I = 1 TO 3
  SET   CNST=I + 1
  SET   DESP[I] = DESP[CNST]
ENDFOR
SET   DESP[CNST] = EREG
```

```
SET    CONT=CONT + 1
IF     SREG = 1
  FOR   I = 1 TO 4
    SET  AUX1 = 0
    SET  CNST=I + 1
    SET  AUX2 = 0
    SET  AUX3 = 0
    SET  AUX1=DESP[I] + POLY[CNST]
    SET  AUX2 = NOT DESP[I]
    SET  AUX3 = NOT POLY[CNST]
    SET  AUX2=AUX2 + AUX3
    SET  DESP[I]=AUX2 * AUX1
  ENDFOR
ELSE
ENDIF
IF     CONT < 14
  GOTO  1
ENDIF
SET    AUX = 0
FOR    I = 1 TO 4
  SET   AUX=AUX + DESP[I]
ENDFOR
IF     AUX = 0
  PRINTLN "Mensaje correcto"
  SET   CORRE = 1
ELSE
  PRINTLN "ERROR: lectura incorrecta"
ENDIF
IF     CORRE = 1
  FOR   I = 1 TO 6
    SET  PETIC[I] = MENSA[I]
  ENDFOR
ENDIF
SET    FIN = 1
END
```

PROGRAM CRCEN

```
*****  
PRINT  "Envio de mensaje"  
SET    FIN = 0  
FOR    I = 1 TO 6  
    SET    MENSA[I] = SITUA[I]  
ENDFOR  
FOR    I = 7 TO 14  
    SET    MENSA[I] = 0  
ENDFOR  
SET    CONT = 0  
FOR    I = 1 TO 4  
    SET    DESP[I] = 0  
ENDFOR  
PRINT  " Polinomio: "  
FOR    I = 1 TO 5  
    PRINT  POLY[I]  
ENDFOR  
LABEL  1  
SET    J=CONT + 1  
SET    EREG = MENSA[J]  
SET    SREG = DESP[1]  
IF     SREG = 1  
    SET    FLAG = 1  
ENDIF  
FOR    I = 1 TO 3  
    SET    CNST=I + 1  
    SET    DESP[I] = DESP[CNST]  
ENDFOR  
SET    DESP[CNST] = EREG  
SET    CONT=CONT + 1  
IF     SREG = 1  
    FOR    I = 1 TO 4  
        SET    AUX1 = 0  
        SET    CNST=I + 1  
        SET    AUX2 = 0  
        SET    AUX3 = 0  
        SET    AUX1=DESP[I] + POLY[CNST]  
        SET    AUX2 = NOT DESP[I]
```

```
SET    AUX3 = NOT POLY[CNST]
SET    AUX2=AUX2 + AUX3
SET    DESP[I]=AUX2 * AUX1
ENDFOR
ELSE
ENDIF
IF     CONT < 10
  GOTO  1
ENDIF
FOR    I = 7 TO 10
  SET   CNST=I - 6
  SET   MENSA[I] = DESP[CNST]
ENDFOR
PRINT  "->Mensaje listo para envio:"
FOR    I = 1 TO 10
  PRINT MENSA[I]
ENDFOR
PRINTLN
FOR    I = 1 TO 10
  IF    MENSA[I] = 1
    SENDCOM 0,49
  ELSE
    IF    MENSA[I] = 0
      SENDCOM 0,48
    ENDIF
  ENDIF
ENDFOR
SENDCOM 0,47
PRINTLN "->completado"
SET    FIN = 1
END
```

PROGRAM ENVAL

```
SET    FINAL = 0
SET    CONT = 1
```

```
FOR    J = 1 TO 12
FOR    I = 1 TO 6
  SET   MENSA[I] = LLENO[CONT]
  SET   CONT=CONT + 1
ENDFOR
SET    FIN = 0
RUN    CRCAL
WAIT   FIN = 1
ENDFOR
SET    FINAL = 1
END
```

PROGRAM METES

```
SET   SITUA[1] = 0
SET   SITUA[2] = 0
SET   SITUA[3] = 1
SET   SITUA[4] = 0
SET   SITUA[5] = 0
SET   SITUA[6] = 1
SPEED 15
OPEN
PRINT  "Solicitud de meter b.sin tratar"
SET   LIBRE = 0
SET   INDEX = 1
FOR    J = 1 TO LEVLS
FOR    I = 1 TO CELLS
  IF    LLENO[INDEX] = 0
    SET   LIBRE = INDEX
  ENDIF
  SET   INDEX=INDEX + 1
ENDFOR
ENDFOR
IF     LIBRE <> 0
  SET   OUT[4] = 1
  MOVED CINT3
  MOVED CINT2
  SPEED 5
```

```
MOVED  CINT1
CLOSE
MOVED  CINT2
SET    SITUA[1] = 0
SET    SITUA[2] = 0
SET    SITUA[3] = 1
SET    SITUA[4] = 1
SET    SITUA[5] = 1
SET    SITUA[6] = 1
SPEED  15
MOVED  CINT3
SET    CNST=LIBRE + 400
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SET    CNST=CNST + 100
SET    LLEN0[LIBRE] = 1
MOVED  CIM[CNST]
SET    CNST=CNST + 100
MOVED  CIM[CNST]
SET    CNST=CNST + 100
MOVED  CIM[CNST]
MOVED  CIM[499]
SET    SITUA[1] = 1
SET    SITUA[2] = 0
SET    SITUA[3] = 0
SET    SITUA[4] = 0
SET    SITUA[5] = 0
SET    SITUA[6] = 0
SET    OUT[4] = 0
ELSE
SET    SITUA[1] = 0
```



```
SET    SITUA[2] = 1
SET    SITUA[3] = 1
SET    SITUA[4] = 1
SET    SITUA[5] = 0
SET    SITUA[6] = 1
DELAY  400
ENDIF
PRINTLN
END

      PROGRAM METED
      *****

SET    SITUA[1] = 0
SET    SITUA[2] = 0
SET    SITUA[3] = 1
SET    SITUA[4] = 0
SET    SITUA[5] = 1
SET    SITUA[6] = 0
SPEED  15
PRINT  "Solicitud de meter b. defect."
OPEN
SET    LIBRE = 0
SET    INDEX = 37
FOR    J = 1 TO 2
  FOR    I = 1 TO CELLS
    IF    LLENO[INDEX] = 0
      SET    LIBRE = INDEX
    ENDIF
    SET    INDEX=INDEX + 1
  ENDFOR
ENDFOR
IF    LIBRE <> 0
  SET    OUT[4] = 1
  MOVED  CINT3
  MOVED  CINT2
  SPEED  5
  MOVED  CINT1
CLOSE
```

```
MOVED  CINT2
SET    SITUA[1] = 0
SET    SITUA[2] = 0
SET    SITUA[3] = 1
SET    SITUA[4] = 1
SET    SITUA[5] = 1
SET    SITUA[6] = 1
SPEED  15
MOVED  CINT3
SET    CNST=LIBRE + 400
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
OPEN
SET    CNST=CNST + 100
SET    LLEN0[LIBRE] = 1
MOVED  CIM[CNST]
SET    CNST=CNST + 100
MOVED  CIM[CNST]
SET    CNST=CNST + 100
MOVED  CIM[CNST]
MOVED  CIM[499]
SET    SITUA[1] = 1
SET    SITUA[2] = 0
SET    SITUA[3] = 0
SET    SITUA[4] = 0
SET    SITUA[5] = 0
SET    SITUA[6] = 0
SET    OUT[4] = 0
ELSE
SET    SITUA[1] = 0
SET    SITUA[2] = 1
SET    SITUA[3] = 1
```

```
SET    SITUA[4] = 1
SET    SITUA[5] = 1
SET    SITUA[6] = 0
DELAY  400
ENDIF
PRINTLN
END

      PROGRAM EXTRS
      *****

SET    SITUA[1] = 0
SET    SITUA[2] = 0
SET    SITUA[3] = 0
SET    SITUA[4] = 0
SET    SITUA[5] = 0
SET    SITUA[6] = 1
PRINTLN
PRINT  "Solicitud de extraccion b. sin tratar"
SPEED  15
OPEN
SET    BAND = 0
SET    INDEX = 1
FOR    I = 1 TO LEVLS
  FOR    J = 1 TO CELLS
    IF    LLENO[INDEX] = 1
      SET    BAND = INDEX
    ENDIF
    SET    INDEX=INDEX + 1
  ENDFOR
ENDFOR
ENDFOR
IF    BAND <> 0
  PRINTLN
  PRINT  "Encontrada bandeja en " BAND
  SET    CNST=400 + BAND
  SET    OUT[4] = 1
  MOVED  CIM[CNST]
  SET    CNST=CNST - 100
  MOVED  CIM[CNST]
```

```
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SPEED  5
SET    CNST=CNST - 100
MOVED  CIM[CNST]
CLOSE
SET    CNST=CNST - 100
SET    LLEN0[CNST] = 0
SET    CNST=CNST + 200
MOVED  CIM[CNST]
SPEED  15
SET    CNST=CNST + 100
MOVED  CIM[CNST]
SET    CNST=CNST + 100
MOVED  CIM[CNST]
MOVED  CINT3
MOVED  CINT2
SPEED  5
MOVED  CINT1
OPEN
MOVED  CINT2
SET    SITUA[1] = 0
SET    SITUA[2] = 0
SET    SITUA[3] = 1
SET    SITUA[4] = 1
SET    SITUA[5] = 1
SET    SITUA[6] = 1
SPEED  15
MOVED  CINT3
MOVED  CIM[499]
PRINTLN "Proceso terminado"
SET    OUT[4] = 0
SET    SITUA[1] = 1
SET    SITUA[2] = 0
SET    SITUA[3] = 0
SET    SITUA[4] = 0
SET    SITUA[5] = 0
```

```
SET    SITUA[6] = 0
ELSE
SET    SITUA[1] = 0
SET    SITUA[2] = 1
SET    SITUA[3] = 0
SET    SITUA[4] = 1
SET    SITUA[5] = 0
SET    SITUA[6] = 1
PRINTLN "ERROR: no hay piezas"
DELAY  400
ENDIF
END

      PROGRAM EXTRD
      *****

SET    SITUA[1] = 0
SET    SITUA[2] = 0
SET    SITUA[3] = 0
SET    SITUA[4] = 0
SET    SITUA[5] = 1
SET    SITUA[6] = 0
PRINTLN
PRINT  "Solicitud de extraccion defect."
SPEED  15
OPEN
SET    BAND = 0
SET    INDEX = 37
FOR    I = 1 TO 2
  FOR    J = 1 TO CELLS
    IF    LLENO[INDEX] = 1
      SET    BAND = INDEX
    ENDIF
    SET    INDEX=INDEX + 1
  ENDFOR
ENDFOR
IF    BAND <> 0
  PRINTLN
  PRINT  "Encontrada bandeja en " BAND
```

```
SET    CNST=400 + BAND
SET    OUT[4] = 1
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SET    CNST=CNST - 100
MOVED  CIM[CNST]
SPEED  5
SET    CNST=CNST - 100
MOVED  CIM[CNST]
CLOSE
SET    CNST=CNST - 100
SET    LLEN0[CNST] = 0
SET    CNST=CNST + 200
MOVED  CIM[CNST]
SPEED  15
SET    CNST=CNST + 100
MOVED  CIM[CNST]
SET    CNST=CNST + 100
MOVED  CIM[CNST]
MOVED  CINT3
MOVED  CINT2
SPEED  5
MOVED  CINT1
OPEN
MOVED  CINT2
SET    SITUA[1] = 0
SET    SITUA[2] = 0
SET    SITUA[3] = 1
SET    SITUA[4] = 1
SET    SITUA[5] = 1
SET    SITUA[6] = 1
SPEED  15
MOVED  CINT3
MOVED  CIM[499]
PRINTLN "Proceso terminado"
SET    OUT[4] = 0
```

```
SET    SITUA[1] = 1
SET    SITUA[2] = 0
SET    SITUA[3] = 0
SET    SITUA[4] = 0
SET    SITUA[5] = 0
SET    SITUA[6] = 0
ELSE
SET    SITUA[1] = 0
SET    SITUA[2] = 1
SET    SITUA[3] = 0
SET    SITUA[4] = 1
SET    SITUA[5] = 1
SET    SITUA[6] = 0
PRINTLN "ERROR: no hay piezas"
ENDIF
DELAY  400
END

      PROGRAM  PROB
      *****

SPEED  10
MOVED  CIM[499]
CLRCOM 0
SET    SITUA[1] = 1
SET    SITUA[2] = 0
SET    SITUA[3] = 0
SET    SITUA[4] = 0
SET    SITUA[5] = 0
SET    SITUA[6] = 0
PRINTLN "Ejecutando inicializacion"
*ANADIR LECTURA ALMACEN*
RUN    INIC
WAIT   PUMAR = 1
LABEL  1
PRINTLN "Esperando mensaje"
SET    FIN = 0
RUN    RECEI
WAIT   FIN = 1
```

```
IF    ERREC = 1
  GOTO  1
ENDIF
SET    FIN = 0
RUN    CRC
WAIT   FIN = 1
IF     CORRE = 1
  PRINTLN
  PRINT  "Petición recibida: "
  FOR    I = 1 TO 6
    PRINT  PETIC[I]
  ENDFOR
  PRINTLN
  IF     PETIC[1] = 0
    ANDIF  SITUA[1] = 1
    IF     PETIC[2] = 0
      ANDIF  PETIC[3] = 0
      IF     PETIC[5] = 0
        ANDIF  PETIC[6] = 0
        PRINTLN "Extraer bandeja tratada"
        SET    FIN = 0
        RUN    CRCEN
        WAIT   FIN = 1
        RUN    EXTRA
      ELSE
        IF     PETIC[5] = 0
          ANDIF  PETIC[6] = 1
          PRINTLN "Extraer bandeja sin procesar"
          SET    FIN = 0
          RUN    CRCEN
          WAIT   FIN = 1
          RUN    EXTRS
        ELSE
          IF     PETIC[5] = 1
            ANDIF  PETIC[6] = 0
            PRINTLN "Extraer bandeja defectuosa"
            SET    FIN = 0
```



```
RUN    CRCEN
WAIT   FIN = 1
RUN    EXTRD
ELSE
    PRINTLN "PETICION SIN COHERENCIA"
ENDIF
ENDIF
ENDIF
ENDIF
IF     PETIC[2] = 0
    ANDIF PETIC[3] = 1
    IF     PETIC[5] = 0
        ANDIF PETIC[6] = 0
        PRINTLN "Meter bandeja tratada"
        SET    FIN = 0
        RUN    CRCEN
        WAIT   FIN = 1
        RUN    METE
    ELSE
        IF     PETIC[5] = 0
            ANDIF PETIC[6] = 1
            PRINTLN "Meter bandeja sin procesar"
            SET    FIN = 0
            RUN    CRCEN
            WAIT   FIN = 1
            RUN    METES
        ELSE
            IF     PETIC[5] = 1
                ANDIF PETIC[6] = 0
                PRINTLN "Meter bandeja defectuosa"
                SET    FIN = 0
                RUN    CRCEN
                WAIT   FIN = 1
                RUN    METED
            ELSE
                PRINTLN "PETICION SIN COHERENCIA"
            ENDIF
        ENDIF
    ENDIF
ENDIF
```

```
ENDIF
ENDIF
IF    PETIC[2] = 1
  ANDIF  PETIC[3] = 0
  IF    PETIC[4] = 0
    ANDIF  PETIC[5] = 0
    ANDIF  PETIC[6] = 1
    SET    SITUA[1] = 0
    SET    SITUA[2] = 1
    SET    SITUA[3] = 0
    SET    SITUA[4] = 0
    SET    SITUA[5] = 0
    SET    SITUA[6] = 1
    PRINTLN "Hacer puesta a punto del almacen"
    SET    FIN = 0
    RUN    CRCEN
    WAIT    FIN = 1
    RUN    INIC
  ELSE
    IF    PETIC[4] = 0
      ANDIF  PETIC[5] = 1
      ANDIF  PETIC[6] = 0
      PRINTLN "Envio completo del almacen"
      DELAY  100
      SET    FINAL = 0
      RUN    ENVAL
      WAIT    FINAL = 1
    ELSE
      PRINTLN "Petición no reconocida"
    ENDIF
  ENDIF
ENDIF
ENDIF
ENDIF
IF    PETIC[1] = 0
  ANDIF  PETIC[2] = 1
  ANDIF  PETIC[3] = 0
  ANDIF  PETIC[4] = 0
```

```
ANDIF  PETIC[5] = 0
ANDIF  PETIC[6] = 0
PRINT  "recibir info del almacen"
SET    FIN = 0
RUN    CRCEN
WAIT   FIN = 1
IF     MENSA[2] = 1
  ANDIF  MENSA[4] = 1
  SET    SITUA[1] = 1
  SET    SITUA[2] = 0
  SET    SITUA[3] = 0
  SET    SITUA[4] = 0
  SET    SITUA[5] = 0
  SET    SITUA[6] = 0
ENDIF
ENDIF
ENDIF
GOTO   1
END
```

```
PROGRAM CRCAL
*****

PRINTLN "CRCAL: "
SET    FIN = 0
FOR    I = 7 TO 14
  SET   MENSA[I] = 0
ENDFOR
SET    CONT = 0
FOR    I = 1 TO 4
  SET   DESP[I] = 0
ENDFOR
LABEL  1
SET    J = CONT + 1
SET    EREG = MENSA[J]
SET    SREG = DESP[1]
IF     SREG = 1
  SET   FLAG = 1
ENDIF
```

```
FOR    I = 1 TO 3
  SET    CNST=I + 1
  SET    DESP[I] = DESP[CNST]
ENDFOR
SET    DESP[CNST] = EREG
SET    CONT=CONT + 1
IF     SREG = 1
  FOR    I = 1 TO 4
    SET    AUX1 = 0
    SET    CNST=I + 1
    SET    AUX2 = 0
    SET    AUX3 = 0
    SET    AUX1=DESP[I] + POLY[CNST]
    SET    AUX2 = NOT DESP[I]
    SET    AUX3 = NOT POLY[CNST]
    SET    AUX2=AUX2 + AUX3
    SET    DESP[I]=AUX2 * AUX1
  ENDFOR
ELSE
ENDIF
IF     CONT < 10
  GOTO    1
ENDIF
FOR    I = 7 TO 10
  SET    CNST=I - 6
  SET    MENSA[I] = DESP[CNST]
ENDFOR
PRINT  "Mensaje->:"
FOR    I = 1 TO 10
  PRINT  MENSA[I]
ENDFOR
FOR    I = 1 TO 10
  IF     MENSA[I] = 1
    SENDCOM 0,49
  ELSE
    IF     MENSA[I] = 0
      SENDCOM 0,48
```

```
ENDIF  
ENDIF  
ENDFOR  
SENDCOM 0,47  
PRINT "->Enviado"  
SET FIN = 1  
END
```


REFERENCIAS

- [1] NPTEL, «NPTEL,» 2013. [En línea]. Available: <http://www.nptel.ac.in/courses/112102011>. [Último acceso: 3 Junio 2016].
- [2] Eshed Robotec , ASRS Manual de usuario, 1996.
- [3] F. Castaño Castaño, *Configuración del autómatas M340 para habilitar el puerto de serie*, Sevilla: Laboratorio de Informática y Robótica Industrial.
- [4] Schneider Electric, Modicon M340 con Unity Pro - Conexión serie (manual de usuario), 2008.
- [5] ESHED ROBOTEC , ATS - Advanced Terminal Software for Controller-B, 1995.
- [6] Schneider Electric, «Modicon M340 para Ethernet - Procesadores y módulos de comunicaciones,» 2009. [En línea]. Available: http://www2.schneider-electric.com/resources/sites/SCHNEIDER_ELECTRIC/content/live/FAQS/86000/FA86427/es_ES/Modicon%20M340%20for%20Ethernet%20Communication%20Modules%20and%20Processors%20Unity%20Pro%2041.pdf. [Último acceso: Junio 2016].
- [7] T. Ramabadran y S. Gaitonde, «A tutorial on CRC computations,» *IEEE Micro*, vol. 8, nº 4, pp. 62-65, 1988.
- [8] R. N. Williams, *A Painless Guide to CRC Error Detection Algorithms*, 1993.
- [9] ESHED ROBOTEC , ACL Advanced control Language - Reference Guide for Controller-A, 1999.
- [10] S. Electric, «Ficha del producto- Conmutador TCSESU083FN0 ConneXium,» [En línea]. Available: http://logi5.xiti.com/go.click?xts=315505&s2=89&p=Industry::2400::584~ConneXium::selector::TCS ESU083FN0_pdf_datasheet&clie=T&type=click&url=http://www.ops-ecat.schneider-electric.com/cut.CatalogueRetrieverServlet/CatalogueRetrieverServlet?fct=get_element&.
- [11] MatrikonOPC, «OPC: ¿De qué se trata, y cómo funciona?,» [En línea]. Available: http://www.infopl.net/files/documentacion/comunicaciones/infopl_net_guia_para_entender_la_tecnologia_opc.pdf. [Último acceso: 01 06 2016].
- [12] F. J. Ramos Ramírez, *Control y supervisión de célula de fabricación flexible*, Sevilla, 2008.
- [13] O. Boix, «Recursos docentes CITCEA,» [En línea]. Available: <http://recursos.citcea.upc.edu/grafcet/gemma/descrip.html>. [Último acceso: 3 Junio 2016].
- [14] A. L. Velasco Martín, *Integración de alimentador de bandejas a célula de fabricación flexible controlada por PLC.*, Sevilla, 2015.

- [15] «Guía de diseño HMI,» de *Sistemas SCADA*, Barcelona, Marcombo, 2007.
- [16] ESPAÑA, «Real Decreto 485/1997, de 14 de abril, sobre disposiciones mínimas,» de *Boletín Oficial del Estado*, 2015, pp. 1-4.
- [17] «IEC 61131-3,» 2013.
- [18] T. V. Ramabadran, «A tutorial on CRC computations,» *IEEE Micro*, vol. 8, n° 4, pp. 62-75, 1988.
- [19] Schneider Electric, «Technical FAQs,» [En línea]. Available: http://www.schneider-electric.com/en/faqs/answeropen/?doc_language=en-US&type=open&searchid=1464549083925&answerid=16777221&iqaction=5&url=http%3A%2F%2Finfocenter.schneider-electric.com%2Fckmic%2Findex%3Fpage%3Dcontent%26id%3DFA198493%26actp%3Dsearch%26vi. [Último acceso: 30 mayo 2016].

